TABLE OF CONTENTS

Chapter 1

Overview of the Finance Standard Client/Server Guidelines

1.1 General

As described in the DMR Group, Inc.'s Network Computing: Strategies for Open Systems, in the DoD Technical Reference Model for Information Management, and in the DoD Technical Architecture Framework for Information Management, clients are defined as hardware, software, persons, or a combination that request services from servers. Servers are defined as hardware, software, or a combination that provides resources to one or more clients.

Client/server is generally characterized by the division of an application into components processed on different networked computers. It is made up of two distinguishable entities: a client, which requests a service or information from a server; and a server, which processes the request, performs the service, and returns the requested information to the client. The client/server model, which is illustrated in figure 1, has the following capabilities:

- The client and the server can interact seamlessly.
- Generally, the client and the server are located on separate platforms and connected via a network.
- The client or the server can be upgraded individually.
- The server can serve multiple clients concurrently and, conversely, a client can access multiple servers.

Applications based on the client/server model of computing provide great latitude in deploying application functionality across a network. This flexibility enables the use of desktop and portable devices, which, when working in conjunction with high-power, low-cost servers, helps provide more cost-effective business solutions. The power of client/server is that it allows support to customers in making the transition from transaction-based processes to business-event and knowledge-based solutions. This transition is accomplished by cost-effective application of new technology enablers such

The Client/Server Model

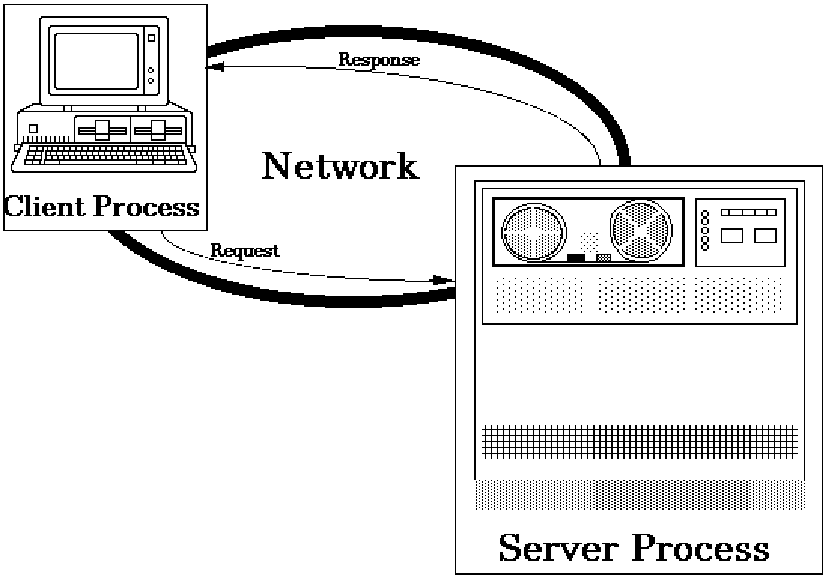

Figure 1

as graphical user interfaces (GUIs), multimedia, and imaging, on an enterprise level. A multimedia application might use audio responses to queries. Imaging provides a means to eliminate duplicative data entry and inefficient paper movement. As image, video, and sound applications become more popular, server system requirements for multimedia storage increase and might include storage of hypertext on optical storage devices and video/sound data on video cassettes, compact disks, and video disks. Client/server also allows movement of data closer to the user by providing both mobile connections to the data and a more cost-effective implementation of relational database technology.

Client/server architecture facilitates the separation of applications into distinct functions: user interface, application logic, and data management and access. This separation allows for major portions of the application to be replaced without affecting the others. This becomes a major portability benefit. Because the portion of application that the user sees, the user interface, is completely isolated from the application logic and data management portions, the application logic or data management code can be updated or completely replaced without the change being visible to the user. Such isolation as provided by client/server architecture makes this level of portability possible.

Standards are a crucial element in client/server architectures, defining the interface between clients and servers and ensuring client application portability, interoperability, and maintainability. The standardization of these interfaces allows client/server communications to operate effectively in an Open Systems Environment (OSE). Two major architectural standards are evolving to provide guidelines and definitions for client/server architectures: the Open Software Foundation's (OSF) Distributed Computing Environment (DCE) and UNIX International's (UI) Open Network Computing (ONC). The DoD Technical Reference Model for Information Management (TRM) describes the standards-based interfaces that are open system targets for the department.

The fact that client/server architectures are not mature must be considered when implementing a client/server system. Organizations are often forced to choose among proprietary approaches. In general, the recommended approach is to define an application and its information requirements as sets of generic components that can be implemented throughout a Network Computing Environment (NCE). The system can then be defined as a set of services and the interfaces between these services and the generic components that require their use. A common approach is to separate the user interface from the application and have it reside on a workstation while application code and databases reside on servers. Chapter 2 of this document describes five basic client/server models.

Four fundamental attributes are associated with a client/server architecture:

- A many-to-many relationship exists between clients and servers. One server can support multiple clients at the same time and one client can access multiple servers at the same time.
- A server can in turn become a client to another server. A server can change roles depending on whether it is providing resources to a client or requesting resources from another server.
- Clients and servers can be, and usually are, replicated through the network.
- Requests for servers are initiated by clients. Servers cannot initiate requests to clients.

1.2 References

1. Strategies for Open Systems, Stage 4, Standards Based Architecture. Chapter 10, Network Computing. These documents are available from the Defense Information Systems Agency, Center for Information Management. Contact Mr. John Keane, DISA-XI, (703) 285-5323.

2. DoD Technical Reference Model for Information Management. This document is available from the Defense Information Systems Agency, Center for Information Management. Contact Mr. John Keane, DISA-XI, (703) 285-5323.

3. DoD Technical Architecture Framework for Information Management. This document is available from the Defense Information Systems Agency, Center for Information Management. Contact Mr. John Keane, DISA-XI, (703) 285-5323.

4. Berkeley Interprocess Communications (BSD 4.2). This document is available from DISA-XM, Technical Integration Services Directorate, Columbus, Ohio. Contact Mr. Gene McManus, (614) 692-5518.

6. Distributed Computing Evolves, *The Sun Observer* Magazine, page 10, September 1992.

7. Client/Server Computing, *Communications of the ACM* Magazine, page 77,
July 1992, Vol. 35, No. 7.

8. Open Systems: Strategy and Standards, Faulkner's Managing Open Systems,
Pennsauken, NJ, Faulkner's Information Services, 1992.

9. Internetworking with TCP/IP, Vol. III, Englewood Cliffs, NJ, Prentice Hall 1993.

10. Client-Server Architecture, New York, NY, McGraw-Hill, 1992.

1.3 Assumptions

In formulating this client/server guidance paper, the following assumptions are made:

- A nonproprietary local area network (LAN)-based architecture supporting the full Transmission Control Protocol/Internetwork Protocol (TCP/IP) protocol and utility suite will be used. That is, end-user workstations will be connected to a LAN server, to each other, and possibly to a corporate data server via a LAN. One or more servers will be attached to the LAN to provide file and print services and X Window client application processes. LANs based upon proprietary network operating systems (NOSs) will not be considered for implementation.

  While the typical departmental LAN server(s) must use a nonproprietary operating system (either POSIX or UNIX), mainframes that use proprietary operating systems (such as IBM compatibles with MVS/XA) must be supported on the LAN as corporate data servers for the LANs and workstations. In addition, LAN-based workstations must be able to establish terminal sessions over the LAN using the TCP/IP TELNET or TN3270 virtual terminal utilities. Most of the mainframes considered in this document are of the IBM S/370 architecture, although there are a few others that have no native support of the TCP/IP protocols. TCP/IP support in these systems will require augmentation of mainframe hardware and software to support TCP/IP protocols and the mounting of file systems on the network.

- Workstations will communicate with the LAN and servers with TCP/IP protocols. The server(s) will be available to provide an Open Systems Interconnection (OSI) gateway service as the need for this is established and gateway software becomes available. While it is technically feasible to put both OSI and TCP/IP into end-user workstations, a dual TCP/IP and OSI protocol stack in each workstation is expensive and difficult to administer because of the large number required. Furthermore, an OSI gateway within each server will provide adequate service.

- A wide area network (WAN) supporting multiple protocols, including TCP/IP, will be available to interconnect remote operating locations, workstations, and, servers.

- A Berkeley sockets-based Application Program Interface (API) will be available for application program development on both the end-user workstations and the servers. A sockets programming library will be available for each platform upon which applications will be run. Suitable languages and compilers for implementing sockets-based APIs will be available on these platforms. A higher-level API will be investigated and made available to further insulate the applications programmer from the underlying API. Reference Chapter 4, Tools, for one high-level client/server API available to DoD components.

Chapter 2

Client/Server Models

2.1 Overview of Client/Server Models

To aid in communication about and understanding of client/server, a high-level model based on the Gartner Group's model of client/server structure has been created. The application is divided into four major functional components:

- Presentation—What the user sees and interacts with. The presentation accepts and validates data entry and displays results. Although a GUI is not inherent in the definition of client/server, in most cases the presentation is in the form of a GUI.

- Application function—Performs data manipulation and calculation.

- Data management—Manages access to data files or databases, coordinates users' requests for data, and ensures data integrity and security.

- Network—Provides the communications medium for client/server interaction. In designing a client/server system, a network component is placed between two of the application components, or one of the application components is divided across the network component.

There are various ways to divide the work of an application between the client and the server. The model applies views of that application to establish client and server responsibilities.

The following table summarizes the functions performed by each of these components.

Component	Functions
Presentation	• Control of keyboard and mouse • Drawing of graphics, fonts, and objects on the screen • Redrawing of overlapping images
Application Logic	• Data manipulation • Report generation • Charting and graphing
Data Management	• Data access • Data integrity • Security • Contention resolution
Network	• Communication between client and server • Error checking • Retransmission of data in error • Formatting data according to network protocol

Views of Client/Server Computing

The following views are depicted in figure 2.

- Distributed presentation
- Remote presentation
- Remote data management
- Distributed function
- Distributed data management
- Distributed process

The views are distinguished by the division and placement of the presentation, application function, and data management components across the various client and server platforms. A component can reside entirely on the server or entirely on the workstation, or it may be split so that part of its work is done on the server and part on the workstation. The decision on how these components are placed is driven by the business needs and operational cost.

The sixth view is distinguished from the first five in that portions of the application and data areas are distributed over the network. Network computing is a goal of client/ server processing. Network computing provides a view in which the network appears as a single massive computer, distributing work and data access as necessary for efficient operation.

Client / Server Views

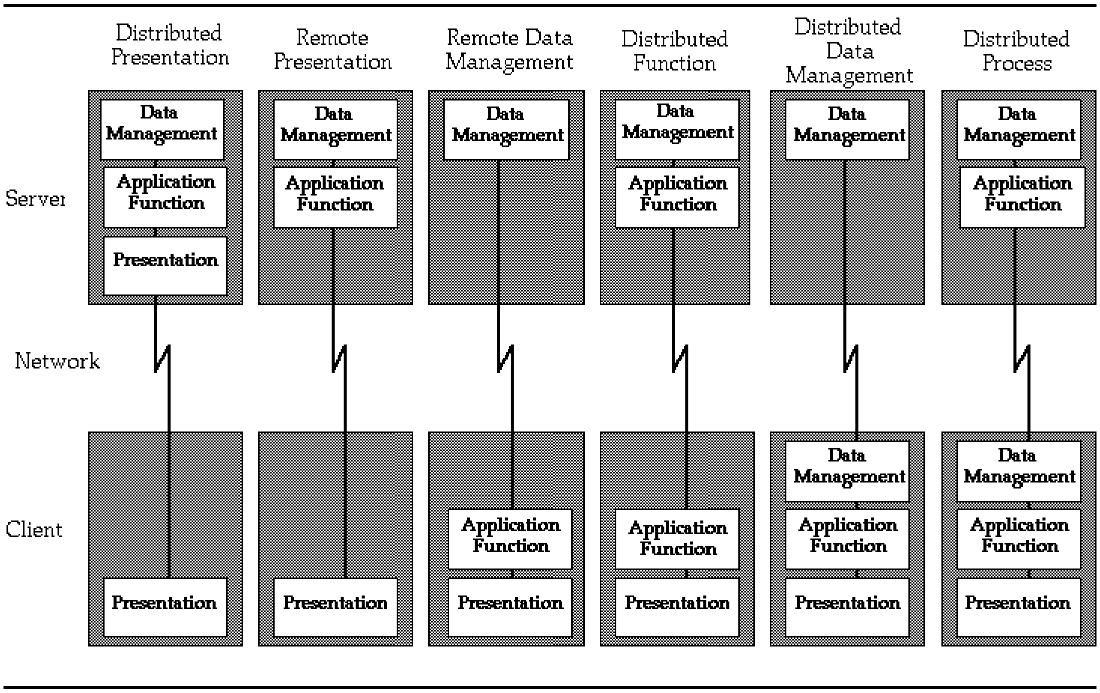

Figure 2

Processing Views

The following table indicates the level of processing power required for the desktop in each processing model.

PROCESSING MODEL	WORKSTATION PROCESSING POWER
Distributed Presentation	Low to medium
Remote Presentation	Low to medium
Remote Data Management	High
Distributed Function	High
Distributed Data Management	Very High
Distributed Process	High—Very High

Client/server can take many forms. The following discusses client/server uses from a mainframe environment perspective. Using this perspective does not imply that the mainframe is always part of the client/server solution, nor is the mainframe being promoted as part of the solution. The mainframe perspective is simply used as a base on which to build a common understanding of client/server.

The following factors must be considered when client/server is implemented:

- Data access security and physical security
- Data integrity
- Data-sharing requirements
- Workstation utilization
- Network requirements

These factors are emphasized not as individual factors, but as an integrated set. They are closely interconnected, and when one factor changes, the others are likely to be affected.

The simplified approach to client/server described here emphasizes off-loading as much processing as possible to the client in order to take advantage of the lower cost computer power of the user's workstation or departmental computer. At the same time, the fact that a mainframe server and a workstation client offer different levels of data integrity and security capabilities is recognized. With these facts in mind, the impact on the network should be considered and the most appropriate of the six views selected for each specific situation.

2.1.1 Classic Client/Server Models

In progressing from the Distributed Presentation Model to the Distributed Data Management Model, the amount of processing performed on the desktop is increasing steadily. The general trend in choosing one model over another is the amount of processing power available on the desktop. For the less powerful desktop computers, Distributed Presentation and Remote Presentation are often the most appropriate choice. Remote Data Management and Distributed Function are often a good selection for high-end desktop computers. Distributed Data Management requires significantly powerful desktop computers.

As the client/server views are presented, there is an implied progression from least difficult as one moves from left to right on figure 2. There is also an implied migration path, which is explained in Section 2.1.3, Implementing Client/Server.

Distributed Presentation

This view of client/server involves the creation of a new user interface for an existing mainframe application. For this reason, distributed presentation is sometimes called beautification. It does not change the mainframe application in any way. Presentation functions are divided between the client and the server (a mainframe). Those functions needed for the mainframe to continue interfacing with 3270 dumb terminals are retained on the mainframe. Those functions that provide a new, user-friendly interface reside on the individual workstations. Thus, workstations can be introduced into the environment while maintaining support for the existing 3270s.

Distributed presentation can be applied to improve the user interface for an existing application. At the same time, this new interface can be integrated with office applications.

Distributed presentation can provide a low-cost, low-risk entry into client/server while providing real benefits to the customer. Therefore, it can be viewed as a potential first step to support a migration strategy to advanced forms of client/server. The Distributed Presentation section provides more information on this view of client/server.

Remote Presentation

Remote presentation is basically the same as distributed presentation; however, under remote presentation, it is no longer necessary to retain the 3270 interface. All presentation functions are moved from the server to the client. The mainframe application then can be modified to provide a more sophisticated technical interface to the workstations. Compared to distributed presentation, remote presentation reduces costs and allows an increase in functionality.

Remote Data Management

In the discussions of distributed and remote presentation, the server was a mainframe and the client was a workstation. In discussing the other client/server views, one needs to remain flexible about what constitutes a server and a client. For example, under remote data management, the server may be a mainframe or it may be a file server serving a local area network. The discussion of the views of client/server will be from the perspective of the server as a mainframe.

With remote data management, data and data access functions are on the server, but application processing and operations on the data are performed on the client (a workstation). All presentation functions reside on the client.

An example of remote data management is the placement of data at an information processing center (IPC) with processing done at connected workstations. This may be done to share data across an enterprise level or when security and data reliability requirements are better met by an IPC-type operation than a distributed-type operation.

Distributed Function

Under distributed function, data resides on the server as it did under remote data management. However, when the functions reside on the client, data must be retrieved from the server, which may impose a significant workload on the network. Distributed function minimizes the workload on the network by placing appropriate application functions on the server to be near the appropriate data. This arrangement decreases the network load but does not necessarily take full advantage of the capabilities of the workstation.

Distributed Data Management

Distributed data management is another way to solve the problem of network load caused by the separation of data and functions. However, instead of placing application functions on the server as is done with distributed function, the appropriate data is moved to the workstation to be near the functions that access it.

The discussions here of both distributed function and distributed data management are simplistic. The methodology for determining the placement of data and processes within a client/server architecture is complex and will be discussed in the Distributed Data Management section.

Distributed Process

Distributed process is the view of the promise of client/server. Although it is technically possible to implement this view today, the state of the technology is such that caution must be exercised in trying to obtain the flexible solutions required.

Under this view, application systems can be designed and implemented so that any of the data or any of the functions can be moved to any component of the configuration. This movement can be done without changing the application, and it may be done as a preplanned operational move or while the application is being executed.

When applied effectively, distributed process takes advantage of the capabilities of the various system components while maintaining a reasonable cost structure by providing flexibility. Flexibility enables changes in at least three possible situations:

- When the presentation, application functions, or data management have been incorrectly placed.
- When the customer's business changes in such a way that there is a shift in how data is accessed and processed.
- When the infrastructure is changed because of things such as technology refreshment or adjustment in the workload due to business changes.

2.1.2 Nonintrusive Processes

"Nonintrusive" processes may not be true client/server processes as described by the classic client/server models. Nonintrusive processes provide a means of accessing information from legacy systems without the requirement for extensive, or in some cases any, modifications to the legacy application. The nonintrusive process provides the entire logic framework necessary to access legacy systems through means provided by the legacy system and/or platform.

- **Nonintrusive Data Access Server**

  The Lawrence Livermore National Laboratory developed the intelligent gateway processor (IGP), commercially marketed as the Control Data Corporation product Ascent, and is an example of a nonintrusive data "server." The IGP/Ascent simulates a terminal session, sending coded keystrokes to navigate through an on-line application, and capturing data from the returned screen data that is read by the IGP/Ascent software and not necessarily displayed to the user. An IGP/Ascent product (or equivalent) may have usefulness as a migration tool. While this implementation offers the possiblity of gaining access to host data with little or no application program changes, the technology is relatively crude and subject to problems such as reading information from a screen position which may change asynchronously to the client, causing incorrect data to be received.

- **3270 "False Front"**

  This is a specific implementation of a nonintrusive "server," which provides a standard user interface, look and feel and/or function key definition to the user, and a back end that simulates a 3270 session to a legacy application. The false-front server may provide a means to simplify and standardize an end user's application view, while minimizing impact upon legacy application code. The server could be a migration tool.

2.1.3 Implementing Client/Server

This section includes an example of a client/server solution for each of the six views. Within each of the six views, hundreds of variations could determine which business functions are placed on which components and what role a given component might fulfill. No single solution is appropriate for every business situation.

The client/server solutions that one chooses to apply depend on the current situation: what specific problems need to be solved, which problems need to be solved first, and what is important. The ultimate goal needs to be determined along with what information technology solutions will support that goal. Once these are determined, several options can be exercised in moving toward that goal.

The most dramatic approach to implementing client/server is to replace an existing application system by developing a new application system for a client/server environment. Although new development is normally a high-risk, high-cost solution, it may be the best solution to meeting one's business needs. However, the experience and interpretation of what is going on in the industry shows that new development is not the prevalent approach.

A more common approach is to migrate an enterprise's application system step by step from a pure monolithic mainframe environment to a client/server environment. This approach leads to maximum use of client/server for the application system or to an eventual total replacement of the monolithic application. Migration can be accomplished in three ways:

- Implement distributed or remote presentation—Refer to the Distributed Presentation and Remote Presentation sections for information and examples of these implementations.

- Improve the application that runs on the mainframe—Use one of the six views to replace a specific mainframe application function or process with a client/server solution.

- Add new client/server applications to complement the application that runs on the mainframe—Put in a new application that solves a business problem or supports a business process that is now cost justified in the client/server environment.

Once any one of these approaches has been accomplished the client/ server infrastructure is in place and can be used to continue to improve business processes. One's migration plan will cause iteration through various combinations of these approaches until business goals are attained. The order that each iteration takes is based on one's business needs and targets of opportunity. There is no prescribed starting point or order of procedure.

The following pages discuss each one of the alternate client/ server views in more detail.

Distributed Presentation

With distributed presentation, the processing performed on the desktop is restricted to very basic user- interface tasks such as painting the screen, gathering keystrokes from the keyboard, detecting mouse movement and button clicks, and sending messages (events) indicating this activity to the remainder of the presentation logic across the network. The workstation may be either graphical or text based. An example of this configuration would be an X Window server (X Terminal). The Army Reserve Component Administration System (RCAS) uses this model to provide presentation services to the end users using the X Window system and protocols.

In the illustration below, the presentation component of an application is divided between two platforms connected by the network component.

```
     Client                                 Server

6444444444444447   644444444444444L4444444444444L4444444444444447
5 Presentation K)))M Presentation * Application *    Data     5
5              5   5              *    Logic    * Management  5
9444444444444448   944444444444444N4444444444444N4444444444444448
```

The following table summarizes the processing that is performed on each of the platforms for an X Window application:

NODE	PROCESSING
A: Display Server Processor on the User's Desktop (X Server)	User input/output (mouse and keyboard)
	Drawing and sizing fonts and graphics images
	Redrawing of overlapped images on the user's screen
B: X Client Process	Telling node A where to place objects for drawing
	Telling node A what the objects look like
	Application logic functions
	Data management functions

The server process in an X Window application is a *display server*, which controls the display and user interface functions of the user's desktop device. This means that *the server runs on the node that would ordinarily be considered the client*. Similarly, X applications are the clients for the display server. These client processes can run on desktop machines or on larger processors normally associated with server processes.

Remote Presentation

In the remote presentation model, the client workstation provides all screen formatting, window management, color processing, and input from keyboard or other input device. The application and server provide no presentation or input logic that is workstation or terminal specific. A protocol between the presentation workstation and the server provides transaction formats that allow input/output (I/O) to take place, and the application to have no need to know workstation specifics. This could be through a standard function call protocol linked with the application. The underlying function call would resolve workstation requirements.

The following illustrates this concept:

```
     Client                                  Server
6444444444444447                  6444444444444L4444444444447
5 Presentation K))))))))))))))))))M Application *    Data    5
5              5                  5    Logic    * Management 5
9444444444444448                  9444444444444N4444444444448
```

The following table summarizes the processing that is performed on each of the platforms for an application using remote presentation:

NODE	PROCESSING
A: Client Application	Local user interface
	User input
	Validation of input
	Shipment of requested transaction to the server (updates, inquiry, etc.)
	Display results of server processing
B: Server Application	Verification of user security access
	Maintenance of data integrity
	Transaction processing (add, change records, generate reports, etc.)
	Transmit results back to client

Remote Data Management

In the remote data management view, the application processing takes place entirely within the workstation. The server provides access to data required by the application. This model would allow distributed processing of centralized data. Using this model, a server is placed into an existing mainframe system to allow access to corporate data by either remote hosts acting as clients or workstations acting as clients. Access to corporate databases is thus accomplished with little or no effect on existing applications, and current applications along with the existing database management system (DBMS) control the integrity of the corporate data. The Defense Logistics Agency (DLA)- developed contract administration system, Mechanization of Contract Administration Services (MOCAS), uses this model in its Contractor Profile System (CPS). Indeed, CPS requests data from some 18 different data servers that are using several different database management systems at the same time.

The following illustrates the remote data management concept:

```
            Client                                        Server
6444444444444444L4444444444444447                       6444444444444447
5 Presentation * Application K)))))))))))))))))))))))M    Data    5
5              *    Logic    5                       5 Management 5
9444444444444444N4444444444444448                       9444444444444448
```

The following table summarizes the processing that is performed on each of the platforms for an application using remote data management:

NODE	PROCESSING

NODE	PROCESSING
A: Client Application	Screen input/output
	Validate data
	Manipulate data
	Create Structured Query Language (SQL) requests
	Transmit requests to server
	Display results from server
B: Server Application	Security
	Data integrity
	Process SQL from client
	Transmit results back to client

An application using the remote data management view could be running in several clients at one time, all accessing the same database. Each type of client workstation would have its own client application. This view allows the server to be upgraded as needed in response to an increased amount of data in the database or an increased number of clients. The server upgrade could be performed without affecting the client applications.

Distributed Function

The distributed function view allows system designers to take advantage of the best available computer resources. It can be difficult and risky to create. Making a poor choice in the division of tasks between the client and server may result in poor response time for the user.

In the distributed function view, the workstation and server split the workload of the application. An example might be that of data entry and validation for the workstation providing input to the main application on the server. The workstation logic would provide all data entry, editing and validation of data, help displays, menus, etc. The data sent to the server would be known to the application there as fully validated and accurate. The DLA Pre- Award Contract System (DPACS) uses this model.

The following illustrates the distributed function concept.

```
              Client                                Server
  644444444444444L44444444444447         6444444444444L4444444444447
  5 Presentation * Application K)))))))))M Application *    Data    5
  5              *    Logic    5         5    Logic    * Management 5
  944444444444444N44444444444448         9444444444444N4444444444448
```

The following table summarizes the processing that is performed on each of the platforms for an application using distributed functionality:

NODE	PROCESSING
A: Client Application	Presents items
	Issues requests to the server
	Receives server responses
	Manipulates results (functional portion) such as reports, statistical analysis, and forecasting
B: Server Application	Receives client requests
	Checks security
	Processes requests (function portion) such as data extraction or manipulation
	Transmits results back to client

Distributed Data Management

The distributed data management view differs from the remote data management view in that each client has a full DBMS and is responsible for any locally stored data. If the data is not resident on the local node, then the client will issue a request for the data to the server. Some commercially available DBMS products currently support this level of functionality.

This model includes workstations that have file systems remotely mounted on one or more servers. A typical scenario is that of almost any workstation attached to a local area network. The workstation has files in use that appear to be on a local disk drive, but that are physically located on the server. The remote location of the workstation data is not apparent to the workstation user. While all proprietary network operating systems provide this service, the Network File System, described in Section 3.3.1, Network Services Servers, presents a nonproprietary approach to distributed data management. The Navy-developed Defense Travel Pay System is an example of an application that uses this client/server model, presently on a proprietary LAN.

The following represents the view of distributed data management:

```
                    Client                                      Server
644444444444444L4444444444444L444444444447          6444444444447
5 Presentation * Application *    Data    K))))))))))M    Data    5
5              *    Logic    * Management 5          5 Management 5
944444444444444N4444444444444N444444444448          9444444444448
```

The following table summarizes the processing that is performed on each of the platforms for an application using distributed data management:

NODE	PROCESSING
A: Client Application	Presentation
	Application logic
	Data management tasks such as data integrity, security, and contention resolution for locally stored data
B: Server Application	Data management tasks such as data integrity, security, and contention resolution for data stored on the server
	Network file system service

Distributed Process

The distributed process view allows application systems to be designed and built so that any data or function can be moved to any component of the configuration. Its flexibility provides the capability to change system components and maintain reasonable cost structures.

The following shows the distributed process view:

```
                Client                                 Server
6444444444444L44444444444L44444444447       64444444444L44444444447
5Presentation*Application*   Data   K)))))))MApplication*   Data   5
5            *   Logic   *Management5       5   Logic   *Management5
9444444444444N44444444444N44444444448       94444444444N44444444448
```

The following table summarizes the processing that is performed on each of the platforms for an application using remote data management:

NODE	PROCESSING
A: Client Application	Presentation
	Application logic
	Data management tasks such as data integrity, security, and contention resolution for locally stored data
B: Server Application	Application logic
	Data management tasks such as data integrity, security, and contention resolution for data stored on the server

Chapter 3

Overview of Client/Server Design

3.1 Selecting Candidates for Client/Server Migration

Perhaps the most appropriate guidance for selecting candidate applications for client/server migration is to "start small."

The development, troubleshooting, and maintenance of client/server applications is complicated by the fact that parts of the code execute in different processes; in many cases different computer systems remotely located from one another. Adding to the complication is the possibility that the computer systems are of differing architectures, with perhaps differing internal data representations that must be accommodated if the application is to run correctly on a given platform.

Careful attention to program structure and program portability is essential to successful client/server implementation. Applications must be designed to be very modular, with program components isolated by general function, such as data entry, data validation, DBMS access, and with each being a distinct and separable module. The interfaces between modules must be well defined.

Data variables must be carefully allocated with portability in mind, so that assumptions made in selecting a variable type that may be different in different computer architectures do not cause applications to fail when the code is ported to another platform. For example, a common assumption which is likely to cause problems is the definition of an integer. Different computer architectures may have differing sizes of hardware registers (8 bit, 16 bit, 32 bit, etc.) This size translates into how big an integer may be without needing to use a second register. Programmers have in the past hardcoded these size restrictions into applications only to have the application fail when it is ported to a platform with a different size of basic register. As a case in point, 80x86-based PCs have 16-bit registers and a typical minicomputer will have 32-bit registers.

Another common pitfall is the way the computer architecture defines the most and least significant in the hardware register. Is the most significant bit on the right or left. This is sometimes known as *endianness*. On some platforms the low order byte (least significant) comes first (little endian); on others the high order byte (most significant) comes first (big endian). If an application is accessing a register in a byte-by-byte fashion as if it were a character array for instance, the program is likely to get the wrong byte first unless these factors are taken into the program design.

3.1.1 Identifying Candidates for Client/Server Migration

A number of influences must be considered when developing a client/server strategy. A strategy is a living, expandable plan that defines the scope of an operation and an approach for applying technology to solve business problems. It must respond to evolving customer requirements and advancing technology.

Not all applications are suitable candidates for migration to a client/server architecture. The decision to migrate a system or application must be made in the setting of the business process the system or application is in. In other words, the decision must make good business sense. In that context there are three categories of systems: those that should be migrated, those that may be migrated, and those that should not be migrated. It is important to always consider the business process in which the system is embedded.

Systems That Should Be Migrated

A system should be migrated if it costs less to move and run the application than to continue in the present environment. For example, if an application has clearly defined boundaries and interfaces, can be made to run faster, and empowers its users at a lower cost, it should be migrated.

Some common features of systems in this category include the following:

- *Is interactive*
  Where users access applications in an interactive manner, client/server processes are in many cases appropriate.

- *Has readily isolated functions*
  Highly structured programs are readily separable into functional components. Program logic that provides data input/output for the application is separate from the application logic. The application logic can be separated easily from the data and user input/output processes.

- *Has well defined function interfaces*
  The function call logic is well defined; functions have only one entry and exit, with a well-defined input and return parameter protocol.

- *Has data structures well separated from the application*
  Data structures are not tightly bound to application logic. Data bases are accessed by SQL calls rather than proprietary DBMS function calls. Ideally, the SQL calls would not be randomly scattered in the application code, but rather placed into a separable set of functions themselves to allow fairly easy modification of proprietary embedded SQL pre-compiler implementations.

- *Is new application development*
  New applications can be designed from the beginning for client/server processes. When designing from the beginning for client/server technology, programs can be structured effectively from the beginning.

Systems That May Be Migrated

A system existing in a business environment that does not support client/server should probably not be migrated even if, when viewed in isolation, it is a perfect candidate for the move. That is, when examined in the context of the business process the system is embedded in the *total* expense would be too great. For example, an application that has clearly defined boundaries and interfaces and can be made to run faster would seem to be a good candidate. However, when viewed in the business context, the added cost of new equipment, personnel retraining, etc., would add to the cost enough to make the conversion prohibitive.

Some common features of systems that could be in this category include the following:

- *Having many remote clients needing access to central data*
  Where data is required for several or many client processes (distributed departmental or personal workstations), it may be appropriate to make server(s) available on the hosts that own and control the data. This is typically more efficient from both a technical and user perspective than downloading a static copy of data to a remote location.

- *Having distributed data accessed by a central application*
  When an aggregation of data that physically resides on a number of different hosts (in perhaps widely dispersed locations), it is appropriate to place server processes in these hosts to allow access to this data while retaining control of the data within the host. The client process can request either a read or update access to the data, with the host(s) managing the access and/or update processes.

- *Is undergoing major change/reengineering*
  Older systems that are undergoing major change and program reengineering may be good candidates. The reengineering may develop a good code structure that will allow client/server processes to be introduced.

Systems That Should Not Be Migrated

Some systems should never be migrated. For these systems, the costs and/or risks far outweigh the benefits. Such a system would fall into one or more of the following categories:

- *Runs in an isolated environment without interaction with other systems or users*

- *Is about to be replaced with a completely new system*

- *Is used infrequently*

Some features common to systems in these categories include the following:

- *Having poorly defined function interfaces*
  If application function interfaces are vague or applied in an unstructured manner, designing for client/server could be difficult.

- *Having data highly integrated into the application*
  Data service is one of the most important client/server implementations. If data usage within an application cannot be separated readily from application logic, accessing data by client/server processes will be very difficult.

- *Is highly stable and not subject to much change*

- *Is not easily subjected to function decomposition*

3.2 Application Clients

Client Characteristics

Applications requesting services from another process, located in the same system or in a network-connected computer system, are clients of those service processes. Examples can range from the very simple (such as the workstation requests for file service for server-mounted shared files used by the Navy-developed Defense Travel Pay System (DTIPS)) to the very complex (such as the MOCAS Contractor Profile System's client departmental computer requesting data from some 18 widely dispersed hosts with a number of different DBMSs). In both cases, some portion of the application is processed in the client computer, and some in the server.

In the case of DTIPS, the entire business system logic of the application is processed within the DTIPS PC workstation, with the workstation and server teaming up to provide data management services for the application. To the end user, the DTIPS application appears to be housed entirely within the PC workstation, and indeed could be if shared files were not needed. DTIPS fits the Distributed Data Management model as described in Chapter 2, Client/Server Models.

MOCAS CPS, on the other hand, puts most of the business system logic in multi-user minicomputers at 14 sites and sends requests for specific contractor data to each of 18 host sites managing four different DBMS products. The CPS application uses a client/server development tool developed by the Defense Logistics Agency called OPENLINK. OPENLINK is described in Chapter 4, Tools. Using OPENLINK allowed CPS application programmers to develop the business system logic for both the client and server, without having to implement relatively complex Berkeley sockets code in the application. The entire effort of opening connections to the hosts, setting up the sockets protocols, and verifying correct data transmissions is handled by the Openlink tool, with only a simple single-function call from the application client.

3.2.1 Workstation Client Systems

Refer to the Workstation Guidelines document for specifics on workstation configuration recommendations.

Personal Computer Workstations

Intel 386 and 486 personal computers can host application client processes in a TCP/IP Open Systems Environment. In addition to the basic PC, a network adapter card and TCP/IP software are required. The Workstation Guidelines document details the components required on the PC that enable it to function as a LAN workstation.

For program development, additional software is required on development PCs. This includes a library of Berkeley sockets function calls, a compiler that is compatible with the sockets library, and perhaps Computer-Aided Software Engineering (CASE) development tools. Also recommended are X Window libraries and a high-level X Window development tool.

UNIX Workstations

Personal computers configured as UNIX workstations, as recommended in the Workstation Guidelines document, all have necessary TCP/IP hardware and software available. These PCs are suitable either as workstations or, with enough memory and disk storage, as low-end departmental servers.

For program development, X Window libraries and a high-level X Window development tool are recommended.

X Terminals

X Terminals are technically servers providing presentation services, but they are included here because they are end-user workstations. These workstations can only be presentation servers, and natively work with TCP/IP LANs. No additional software is required to use an X Terminal. The application running as an *X Client* on another computer must be written to use X Window protocols in order for an end user to be able to use an X Terminal. In the early migration, X Terminal usability will be very limited, as only a very few applications are currently in development as X Window applications. One such application is the Functional Development Maintenance System (FDMS), a portion of the Defense Civilian Personnel Data System (DCPDS) being developed by the Air Force Military Personnel Center at Randolph Air Force Base.

3.2.2 Departmental and Corporate Host Client Systems

Departmental minicomputers and DoD corporate host systems, normally used as servers, can also be clients for other services. A system can at one instant be a server, providing some service to another, and at the next become a client, requiring the services of one or more other systems.

The hardware and software that allows a departmental or corporate system to be a server also allows it to be a client of other servers.

3.3 Servers

Server Characteristics

In general, a server should be *suited* to the role it is called on to perform. The server needs to fit the business process as well as the application environment. Suitability considerations include choosing a compute server with enough central processing unit (CPU) horsepower or a file server with enough disk space. If the server is to be a database engine, then it should have the compute power to manage the database and the I/O power to handle the user requests for data access. If it is to be a communications server, then it should have the connection capacity to service the user community connectivity needs. In all cases, it should be well connected to the LAN so that the seam is not a bottleneck.

A server should be *matched* to its clients. It should have an architecture that allows ready interoperation with its clients. The server must support the identical protocol suite supported by the client in order for them to work together. Also, servers generally support multiple clients, and are normally either as powerful or more powerful than the client computer systems. For example, it probably does not make sense for a microcomputer to be a server for a mainframe.

Two general types of server processes are outlined below. The first, *Network Services Servers*, provide services that are not application specific, but available for use by end users, the network, applications, or other processes that require these standard services. These services are required for overall system management, security, various file transfer utilities, and general presentation services. The second, *Application Servers*, are those that are developed specifically to support one or more application functions, such as database access.

3.3.1 Network Services Servers

As technology continues to evolve, server specialization is extending beyond that of a database server to such functions as communications, terminal emulation, fax, library management, and electronic mail. The following functions will be required of such servers:

1. File Sharing Service

A primary network service is the ability to share files between LAN-connected workstations. Network file sharing service will be provided on LAN-based servers by the Network File System (NFS). NFS will be replaced in time with the Distributed File System (DFS), which is provided by the Open Software Foundation (OSF) Distributed Computing Environment (DCE), detailed below.

Clients may have a need to share the same data file in a workgroup environment. This data file is then placed on a file server (a shared file processor) and clients send their I/O requests to the file server. A file server generally provides a client with access to the entire file, so that if one client updates a shared file, no other clients are able to access the file during that client's access.

Server file system(s) will be "mounted" on the users' workstations during the workstation boot process. These file systems will access files in the workstations' native form, transparently to the user. The files will appear to be on the users' workstations, but will physically reside on the server(s). In this manner, files may be shared among several users on a LAN. File and/or record locking will be done on the server to ensure data integrity.

2. Network Printer Sharing Service

In a networked work area, there is little need for individual printers attached to each workstation. This is especially true of the relatively expensive laser printers. The server will manage one or more printers in a pool, available to all users connected to the LAN using the UNIX/POSIX line printer server (LPD) with the matching printer client (LPR) on the workstations. Network printers will be available for workstation use just as if they were physically attached to the workstation.

One high-quality, high-capacity printer may replace all individual client printers in a workgroup environment. All clients may send file print requests to a print server. A print server maintains a queue of all files to be printed, sending each print file, in turn, to a shared printer. Individual print files are typically printed with a special separator page indicating the client name and file name.

3. Virtual Terminal Service

Workstations connected to a LAN will still need occasional connection to hosts located on the network. The standard TCP/IP TELNET service will be provided to allow a "virtual" terminal on the users' workstations.

TELNET is an example of the TCP/IP remote access application, a virtual terminal facility that allows a user to connect to a remote system as though the user's terminal is physically attached to the remote system. The TELNET protocol defines interactive, character-oriented communication. This protocol specifies a network virtual terminal (NVT) that consists of a keyboard and display screen. The primary advantage of using a network virtual terminal is that it permits clients from a variety of computers to connect to a service.

From the user's point of view, TELNET converts the user's terminal to a terminal connected directly to the remote system. The remote system displays a prompt that requests the user to enter a login identifier and a password when the user invokes TELNET.

This virtual terminal service will be extended to include 3270 terminal connection to IBM-compatible mainframes by including a TCP/IP component for the mainframe host and a 3270 virtual terminal for the users' workstations. The TN3270 product will form the basis of the virtual 3270 workstation software.

4. Network News Service

An important service to be provided to workstation users, as well as others in the

DoD, is access to the network news. The network news provides worldwide and organization-wide information service to connected sites. News articles are grouped by general subject area, such as dod.general, dod.announce, and many other categories that may or may not be DoD or Component related. DoD newsgroups will contain important DoD and/or CIM documentation, such as the HCI Style Guide, or CIM Technical Reference Model in electronic form, available for download world wide. It can also be used to post job announcements, policy notices, or any other information. The TCP/IP Network News Transfer Protocol (NNTP) server will be used to provide client/server-based network news to the workstations.

5. File Transfer Service

Worldwide file transfer will be available from all DoD LAN-connected workstations. This facility will be provided by the TCP/IP File Transfer Protocol (FTP). An FTP client will be on the workstation to communicate with any FTP server in the DoD.

FTP can operate either with known, authorized users authenticated by the FTP server when the service is requested, or "anonymously" where the user need not be known to the server in order to retrieve nonsensitive files. While anonymous FTP can be very useful, it does present some system security challenges and must be carefully administered. It will be recommended that anonymous FTP servers be limited to installations that can commit to an increased level of system administration skill and activity. The typical office LAN server should not support anonymous FTP, but rather require authentication in order to send or receive files. As OSI file transfer (FTAM) components become available, this service will be included in the LAN server via an application gateway.

6. Electronic Mail Service

Another important network service is that of electronic mail (E-Mail) management. The TCP/IP Simple Mail Transfer Protocol (SMTP), coupled with the UNIX/POSIX sendmail service and Post Office Protocol (POP) service will be used to provide worldwide, standards-based electronic mail service. Each workstation will interoperate with the LAN server via a POP client process called a "User Agent." PC workstation software for POP mail user agents is available in a number of configurations to match user needs. This ranges from simple, freeware POP mail clients that operate under MS-DOS without Windows, to relatively elaborate Window-based integrated TCP/IP packages that include electronic mail user agents.

As OSI electronic mail (X400) components become more widely available, this service will be included in the LAN servers as a gateway service.

Privacy Enhanced Mail (PEM) is being investigated as an additional feature that will allow authenticated mail service.

7. Security Service

Enforced security services are required to ensure that user authentication is supported as well as authorization for client access to databases, to specific tables in the databases, to network services, and to the applications that the client is approved to access.

A method of providing for authentication of users and issuing a time-limited authorization for selected services will be provided. The authentication service will be provided by the Kerberos server. Kerberos was developed as a part of the Massachusetts Institute of Technology (MIT) project Athena, a very large network computing installation based upon UNIX servers.

Once authenticated by Kerberos, a user can access any data and/or processes for which he/she is authorized without having to do any further login or password validation, so long as the data or process server interoperates with Kerberos.

Trusted Kerberos authentication servers will be located in physically secure facilities, typically information processing centers (IPCs). Because the Kerberos servers authenticate users and validate requests for access to mission-critical data and processes, they must themselves be secured and administered by skilled, responsible security administrators. Kerberos servers should be located regionally, with several redundant backup servers available in case of outage.

Kerberos is a part of the OSF DCE, described below.

8. Standard Time Service

Within TCP/IP, a service is available that allows one machine to obtain the date and time of day from another machine. A program executing on a client sends a request to an executing server program. The server obtains the current date and time of day from an authorized external service, encodes the data in a standard format, and returns it back to the client. This information is required by other network services and is used to schedule and control processing. Network latency can be a problem. However, a delay approximation can be made and added to the time of day that the server returns to the client.

A known, accurate time is required by the Kerberos security service and other functions. Because there can be wide variance between clocks in computer systems, a means to synchronize system clocks within some small tolerance is required. Standard time service should be regional, with several redundant backup time servers available in case of outage.

9. Domain Name Service

Advanced naming or directory services are available and running on networks today that provide a mechanism through which servers resident on that machine can be located. To field requests of a specific type, servers indicate their availability. When a client request is made, and if a service is available, a naming service returns the address to which requests can be sent. When converting from a domain name in text form, the BSD UNIX socket interface provides library routines to perform conversions to an IP address. At that time the client may contact the server directly. The naming service address could be a well-known port number, defined at boot-time or retrieved via broadcast polling techniques.

A name service can be provided by any UNIX/POSIX system that supports the Domain Name System (DNS). DNS servers provide a means to look up network addresses for computer systems on the network. The domain name service makes it unnecessary for users to know the specific Internet address of a given system. Coupled with a standardized naming scheme, merely knowing the component and perhaps base or station allows an automatic network service to provide message traffic to the correct host. The UNIX DNS will be used for this service.

10. Directory Service

To describe, record, and find characteristics of various services and information they provide, networks require names and directories. An electronic mail system must be able to locate a user's mailbox in order to deliver the mail. Directories can be organized into hierarchies in which a directory can contain other directories.

An OSI X500 directory service will be provided to provide a means to locate computer systems, files, individuals, electronic mail addresses, and many other look-up requirements. The first implementations of the X500 directory service will run over a TCP/IP protocol suite, using the freeware ISODE product. This will allow the development and early use of OSI X500 directory services in advance of full OSI implementations around the department. This directory could form a mechanism for enabling cross-functional data sharing by placing appropriate information in one or more global X500 directories. Workstations will be provided access to this service via a special LAN server implementation of the "who is" or "finger" TCP/IP services.

11. Presentation Service

In the case of Distributed Presentation, the presentation part of the application code is split between two or more network nodes. Therefore, a portion of the presentation services will reside on and be provided by the server. Normally, the Distributed Presentation model will include a front-end component and a back-end component. The front-end presentation component resides on the workstation and handles the physical part of the user interface to include the following:

- Screen displays
- Graphical user interfaces
- Window management
- Color
- Fonts
- Mouse
- Keyboard

The back-end presentation resides on the server and performs common, shared presentation functions.

The X Window System for UNIX-based platforms, Easel and Infront graphical user interfaces for DOS and OS/2, and OS/2 Presentation Manager are examples of Distributed Presentation implementation. When used in combination with PCs to deliver graphical user interfaces to mainframe applications, Distributed Presentation is especially beneficial. This is one way to leverage existing investments in host applications, databases, PCs, and LANs.

A standard presentation service will be provided for DoD applications in the form of the X11 X Window protocols system. X Window, coupled with the MOTIF Graphical User Interface will provide the standard graphic desktop environment for DoD applications. X Window server software will be provided on workstations, and client software will be provided on application server platforms. In the X Window system, the workstation is the server because it provides *presentation services*, unlike other client/server processes.

GUI programming can be cumbersome, time consuming, and error-prone, and can distract from the major task of developing applications. It will be necessary to provide X Window GUI builder software for application developers. A GUI builder allows the screen layouts to be done at a high level, such as a screen painter, generating source code for the GUI in C or Ada.

12. OSI Gateway Service

A number of the network services described above are based upon TCP/IP standards, rather than OSI. In order to provide OSI transition, while simplifying workstations as much as possible and keeping workstation software costs to a minimum, an OSI gateway service will be provided on each LAN server. This will include an electronic mail SMTP-X400 gateway, FTP-FTAM file transfer gateway, and others as necessary. It is intended that OSI gateway services be provided early in the LAN implementation process. OSI gateway products are available on a number of different platforms and DoD contracts, making the service easy to obtain.

3.3.2 The DCE Client/Server Architecture

Open systems and distributed cooperative processing are key to the client/server architecture. One of the best known candidates to become a de facto standard is OSF's Distributed Computing Environment. It meets or exceeds the DISA Technical Reference Model for Information Management's specifications and consists of the following integrated components (See figure 3-1):

- Distributed File System
- Directory Service
- Remote Procedure Calls
- Threads Services
- Time Services

1. Distributed File System (DFS)

Distributed file systems permit a user on one system (that is connected to a network) to access and modify data stored on files in another system. File server data is accessed by copying that data and storing (*caching*) it on the client system to allow the client to read and modify it. Modified data is then written back to the server. A problem exists when more than one client tries to access and modify the same data.

DCE DFS forces the file server to keep track of each client and its cached data. It uses *tokens* to keep track of cached data. Based upon the type of access the client requests, tokens are distributed to a client by a server when the client caches the data. If a client wishes to modify data, it must request a *write token* from the server. If a write token has been allocated, the server informs other clients that the write token has been issued. Other clients will also receive a message that their data is no longer current and the server revokes the token.

DCE DFS features include the following:

- *Access security and protection*. DFS supports both *authentication* (Kerberos system) and an access control list method for granting file access to authorized clients.
- *Data reliability*. DCE DFS supports *replication* for its network services to ensure that any single point of failure does not result in the client being unable to continue processing. If one server

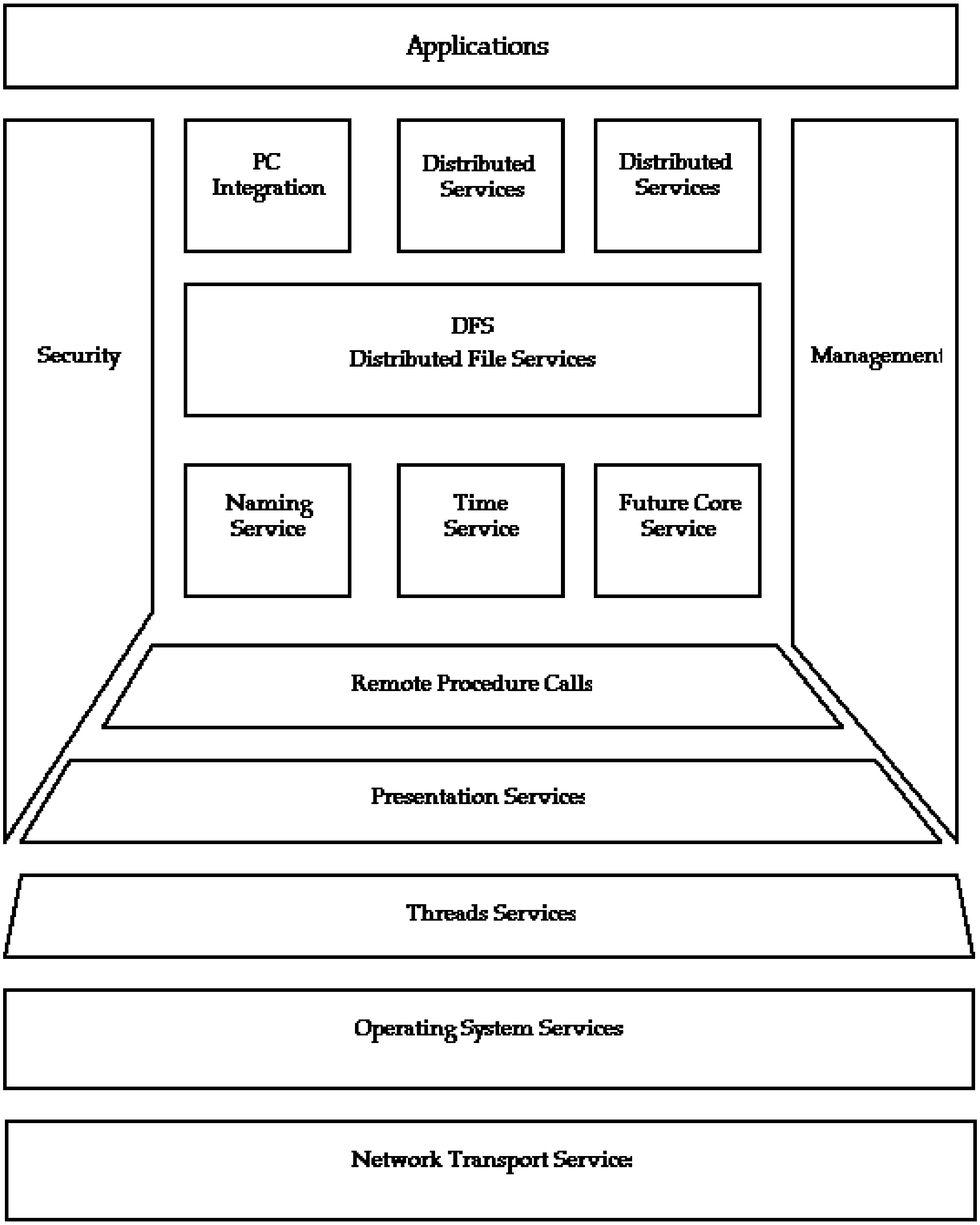

Figure 3-1 DCE Architect

becomes unavailable, the client is automatically switched over to one of the replicated servers.

- *Data availability*. The system administrator is allowed to perform maintenance (file movement, data backup, etc.) on network resources without bringing down any servers or the network.

- *Performance*. DCE DFS is efficient and capable of being extended. It caches file status data on a client system, which reduces the number of client data requests and thereby reduces the server and network workload.

- *Manageability*. DFS uses distributed databases to keep track of file location, authentication, and the access control lists used by the clients and servers. The domains of these databases are separately administered and maintained and can be accessed by any client.

- *Standards conformance*. DCE DFS conforms with the IEEE POSIX 1003.1 file system semantics standard.

- *Interoperability with Network File System (NFS)*. DFS provides a migration path from NFS to DCE DFS by providing gateways allowing clients using NFS to interoperate with DCE DFS.

DCE DFS is based on the Andrew File System from Transarc Corporation. It differs from Sun's Network File System (a current de facto standard) in two main categories:

- DFS uses global file space allowing all network users to see the same paths to accessible files. In NFS, each network node has a different view of the file space. Global file names ensure uniform file access from any network node via a *uniform name space*.

- NFS was designed primarily to operate in a local area network environment. DFS provides integrated support for both local and wide area networks.

2. Directory Service

Anything that can be named and accessed individually (network services, electronic mailboxes, computers, etc.) is called an *object* in DCE. Each object has a corresponding *entry* in the directory services. Each entry contains attributes that describe the object. The name entries are collected into lists called *directories*.

DCE objects are defined by their names, and applications and services gain access to objects by accessing the appropriate directory entry and retrieving its attributes. This indicates the importance of the name and directory services to the DCE. Object characteristics are separated from the object itself and, most important, the *location independence* of objects is assured. This type of organization allows applications and services to access objects even if the object moves or changes some of its attributes.

The DCE Directory Service is integrated with the other DCE components, including DCE DFS, and possesses the same characteristics (security, reliability, availability, manageability, and performance) as the DCE DSF.

Designed to participate in OSI's X500 worldwide directory service, local DCE users can be tied into X500 directory service. In addition, users in other parts of the world are allowed to access local names via X500. DCE supplies naming gateways called Global Directory Agency (GDA). For example: A local client in one part of the DCE network that needs to look up the name of a remote client sends its request to a local GDA residing on a name server. The GDA on that server forwards the request to the worldwide X500 service, which looks up a name and returns the result to the GDA, which in turn passes it back to its client.

The Open Software Foundation DCE uses a service- independent Application Program Interface to ensure portability and interoperability, and to isolate application programmers from the details of the underlying services. This API is based on the X/Open Directory Services (XDS) API specification. Applications using XDS can work with the DCE Directory Service and with X500 without modifications.

3. Remote Procedure Calls (RPCs)

An extension of high- level language subroutine calls is represented by the syntax, semantics, and presentation services of the RPC. RPCs allow the actual code of the called procedure to reside and be executed on a physically remote processor transparent to the application. RPC is the most critical segment of the entire DCE architecture. DCEs RPCs are easy to use, and designed to be transparent to various network architectures and to support threads as described below. RPCs syntax, semantics, and presentation services are the primary differences between OSF's DCE and UNIX International's (UI's) Open Network Computing (ONC). In addition, OSF disagrees with the way ONC encourages fundamental RPC protocol modifications by users.

4. Threads Services

Network environments typically achieve their goals by linking participating processors, providing opportunities to implement some degree of parallel processing. OSF selected the threads strategy for its DCE. This strategy uses subprocesses (threads) that exist and operate within a single instance of the executing program and its address space. The program itself can use special synchronization tools, such as semaphores (flags or monitors), to control access to a common, modifiable resource shared by several users (for example, a memory variable).

Shared memory among multiple programs or the use of explicit synchronization verbs to exchange messages among programs are examples of other methods that can be used to implement parallel processing. However, these methods usually involve resources external to the program. DCEs Threads Services (based on DECs Concert Multithread Architecture) offers portability and supports the POSIX 1003 application and system services interface specification.

5. Time Services

OSF selected DECs distributed Time Synchronization Service for its completeness and simplicity. The function of the time services component is to synchronize the clocks of all network nodes with the clock that controls the network as a whole.

Because OSF DCE was designed to fit into the client/server model, DCE components must be present on the service requestor (DCE client) and the service provider (DCE server) as indicated in figure 3-2. It is not simply a software package that can be installed on a server. DCEs components are placed "between" applications and networking services on both the client and the server. The DCE client/server model hides actual details of services from end users even though DCE is a multi-layered architecture containing a number of basic services.

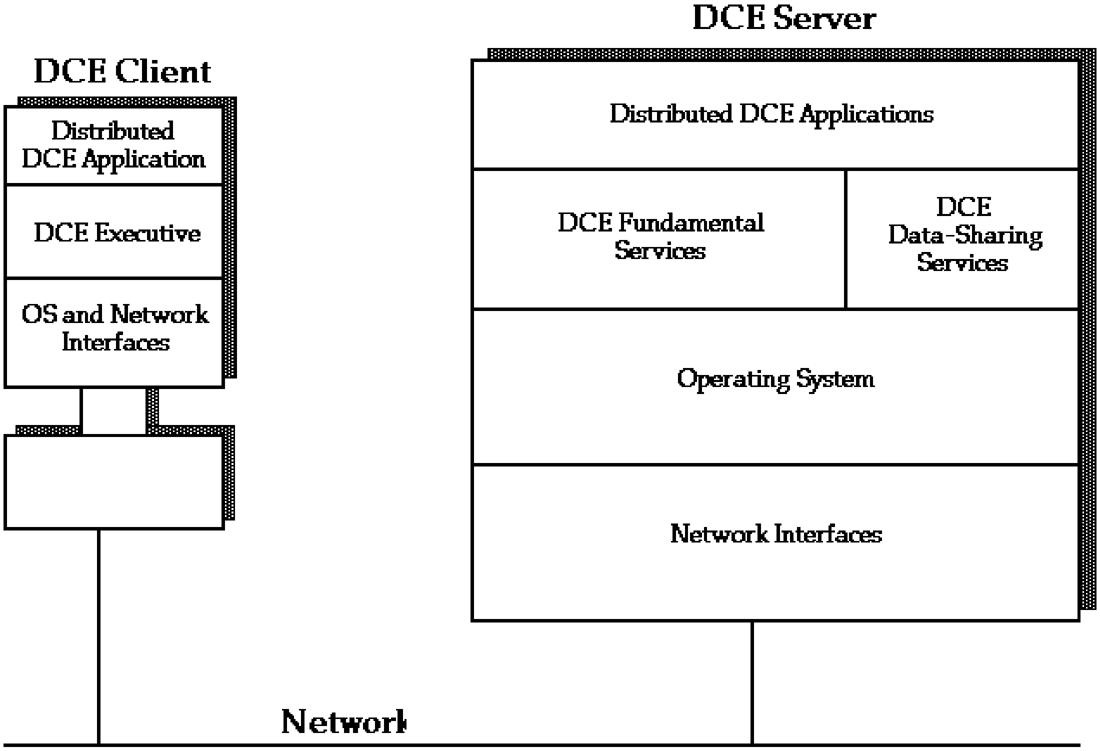
DCE Server
DCE Client
Distributed DCE Application
DCE Executive
OS and Network Interfaces
Distributed DCE Applications
DCE Fundamental Services
DCE Data-Sharing Services
Operating System
Network Interfaces
Networl

Figure 3-2 DCE Client/Server m

3.3.3 Application Servers

1. IBM-Compatible Mainframe Data Servers

A. General

A very high percentage of computer systems that manage DoD corporate data are mainframes, mostly IBM S/370 compatibles running the proprietary IBM MVS operating system (primarily MVS/XA). The natural communications protocols in this environment are not the open TCP/IP protocols necessary for client/server processing as described in this paper, but rather proprietary SNA protocols requiring SNA communications front-end processors (FEPs).

In order to implement servers in the mainframe hosts to supply data on demand to clients using open protocols, and perhaps some application processing as well, it is necessary to augment these hosts with TCP/IP components. This augmentation includes software, the TCP/IP drivers through which applications communicate with the underlying protocols, and hardware for connection to a LAN local to the host's IPC. The IBM-compatible mainframe server runs as either a started task or MVS subsystem. The following diagram illustrates the requirement.

```
644444444444444444444444444444444447
5                                   5
K))))))))))))))))))))))),            5
5     Application     *              5
5                     *             5
K)))))))))))))))))0))))2))))))))))))M
5    TCP/IP      *      VTAM        5
5                 *                  5
9444L4444444444444N444444444;44444448
+)))2))),                  5
*TCP/IP *                  5
*Adapter*            +))))J)))),
.)))0)))-            *   FEP   *
))))2))))))))))))Q     .))))))))))-
Local Area Network        To SNA
(Ethernet)          Network
```

Note that the implementation of TCP/IP software and hardware in the host in no way impacts its ability to continue to process data and communicate over the SNA links. The TCP/IP components provide true added value.

B. JCALS Global Data Management System (GDMS)

A promising technology that is emerging from the Joint Computer-aided Acquisition and Logistics System (JCALS) is the Data Management (DM) subsystem, which includes GDMS. The DM subsystem executes as three cooperating processes hosted on the JCALS Data Management Processor (DMP). The general process flow shown in figure 3-3 is centered around the GDMS. These processes communicate via UNIX domain sockets. The Local Data Management Service (LDMS) will fork and execute processes for each transaction needed with any one DBMS. The GDMS will communicate with other GDMS servers executing at other sites.

Once the GDMS server is activated, it accepts socket connections from client applications and other GDMS servers. Each socket connection represents either a user transaction or an external GDMS sub-transaction. The GDMS will query the Dictionary-Directory module for location information, priorities, data triggers, constraints, alerter, and security based information. The user request is then localized at a global level; that is, the GDMS locates the site(s) that contain the information. GDMS sub-transactions are created, scheduled, and dispatched to either the LDMS running locally or to other GDMS servers executing at remote sites. GDMS provides for data integrity via a Global Data Integrity component that includes two-phased commit logic.

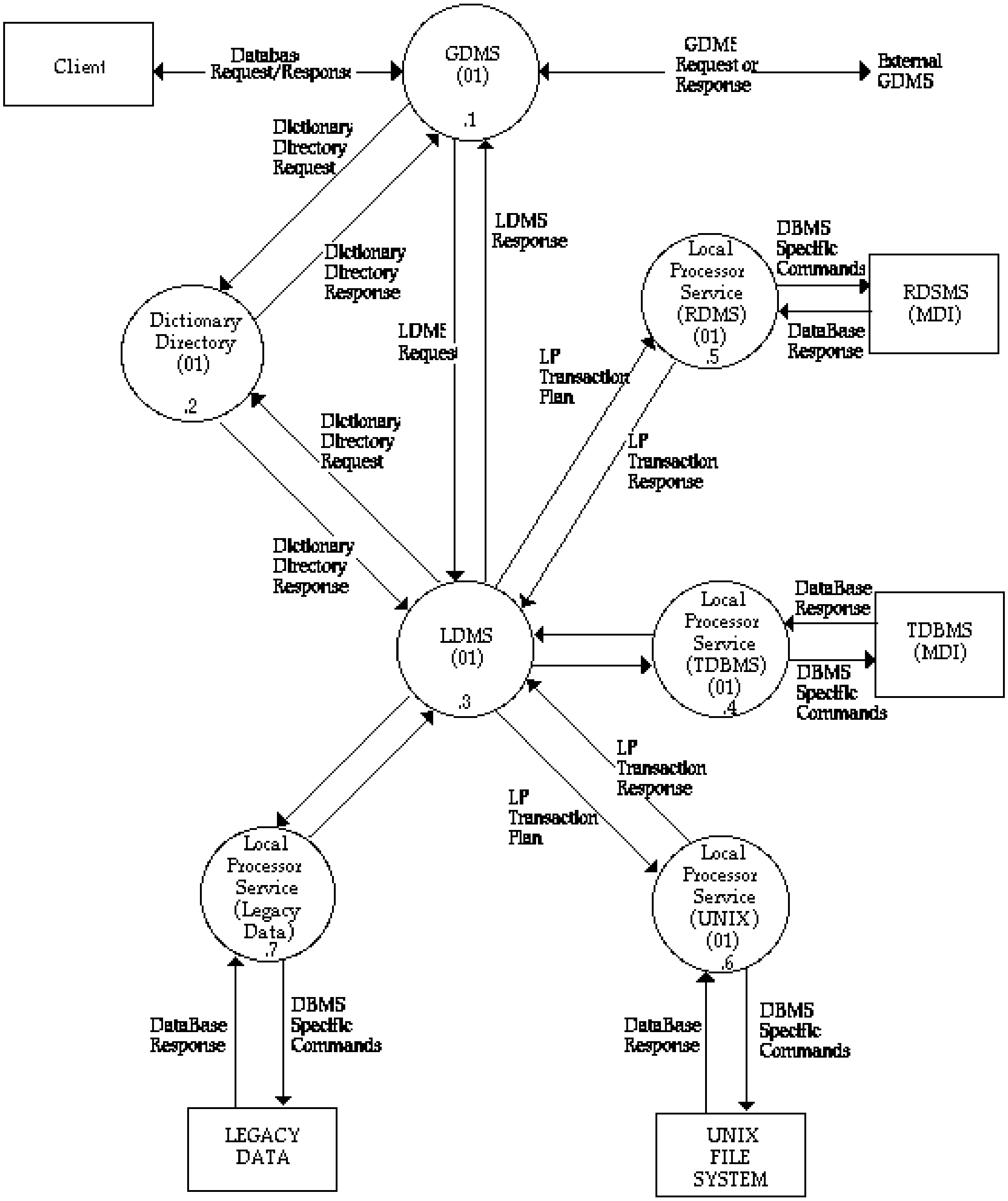

Figure 3-3 DM Process Fl

The LDMS will query the Dictionary-Directory module for local location information. The GDMS request is then locally localized; that is, the LDMS determines the DBMS executing on the local processor that contains the requested data. Local sub-transactions are created, scheduled, and dispatched to the various modules responsible for managing connections and transactions with these DBMSs. The LDMS assembles the local sub-transactions responses into a single LDMS response. The GDMS in turn assembles the LDMS response with any GDMS responses and returns the information to the client application.

The JCALS DM system is currently under prototype development by the JCALS contractor, Computer Sciences Corporation. JCALS will go through the MAISRC II process is early 1993. While the GDMS component is *not* currently available, the functionality to be provided by the JCALS DM system will bear close watching as a relatively near-term data server available within DoD.

2. "Departmental" Minicomputer Servers

In the departmental server environments (UNIX and/or POSIX), TCP/IP hardware and software is common and very inexpensive. All departmental servers recommended for DoD Automated Information System (AIS) implementation will be specified to include the required TCP/IP hardware and software as a standard purchase.

3. Layering Server Code for DBMS and TCP/IP Isolation

Application Program Functions

In its simplest representation, an application program can be thought of in three basic processing layers representing application functions summarized below and depicted in figure 3-4:

- *Presentation logic*. Presentation logic performs screen formatting, dialog management, reading and writing of the screen information, window management, and keyboard and mouse handling tasks. This part of the application code interacts with the end user's terminal. Advanced presentation logic can handle such functions as data type and range validation, cross-field editing, context sensitive help, message logging, and access control.

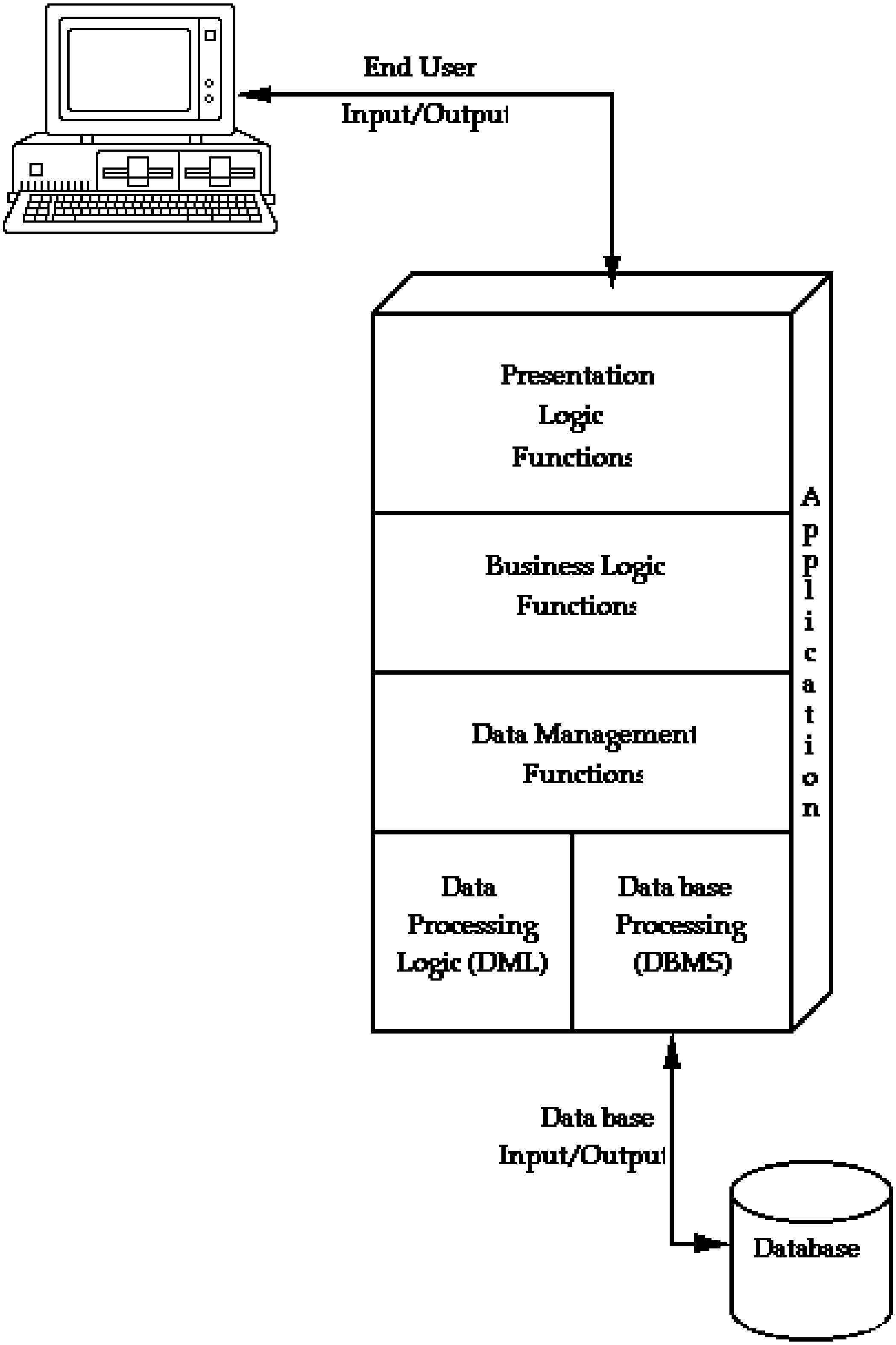

Figure 3-4 Typical Application Compon

- *Business processing logic*. Business logic processes the input data according to the requirements, rules, and algorithms of a particular business task it is designed to perform. This part of the code is not directly related to end-user and/or database I/O. Usually, business processing logic is user written in any of the supported third-generation (3GL) languages (e.g., COBOL, C, PL/1), or in higher level fourth-generation (4GL) languages (user written or generated by a code generator).

- *Data management logic* consists of two components:

  - *Data processing logic* - A part of the application code that manipulates data within the application. Typically, a data manipulation language (DML) is embedded into the 3GL or 4GL application code. Data residing in the relational database managed by the relational DBMSs (RDBMSs) is accessed using some dialect of SQL. The use of embedded SQL will be discussed in more detail below.

  - *Database processing* - The processing of the data that relates directly to the requests prepared in the DML (physical I/O, buffer, log and lock management, etc.). Performed by the DBMS, this low-level data management is hidden from the business logic of the application. However, within the context of the architecture, data processing is an essential part of the application logic.

Each of the layers could be further decomposed, which in general is a good design and development approach.

Properly designed, an application program provides a strong separation between these three layers, to the point that each could be, and in many cases is, a completely separate section of code, developed and debugged entirely separately from the others. A well-defined interface to each layer provides a standard method of communicating parameters and data from one layer to another. These layers are typically individual modules, called functions or procedures, which are called by other functions requesting the services of the layer. The function call interface describes how each function is called, what parameters are required, and the type of data each parameter must be.

Presentation logic is wholly responsible for displaying information for the end user on a workstation; interactions between presentation logic, business processing logic, and data management logic are performed in a cooperative way. These interactions take the form of clients' requests and servers' responses to those requests.

Business processing logic is wholly responsible for the processing required by the business system. This is the heart of the application providing business area support to the functional user, and it is the part that the applications programmer must be most concerned about.

Data management logic is wholly responsible for accessing data, either flat files or formal databases, on the data host.

Many RDBMSs provide the application programmer a means of embedding SQL statements in the application's source code. In most cases, this embedded SQL feature is implemented in a way proprietary to the DBMS vendor. The DBMS vendor provides a *pre-compiler* that reads the application source files prior to the language compiler and turns the embedded SQL code into proprietary function calls to the RDBMS. It is important to strongly separate embedded SQL statements into a section of code that can be readily modified to allow porting the application to another vendor's RDBMS. The ideal application program interface would be a function that is passed a character string representation of the SQL statement. This function could then perhaps have the proprietary precompiler code necessary to access the RDBMS. When porting the application to another RDBMS, the application's *business processing logic* would not require modification.

Applications developed with structured programming techniques and proper software engineering guidelines can help separate these functions. However, it must be recognized that separation of application logic into these categories is not always straightforward in complex data systems, and boundaries between the functions are not always clearly defined.

In a typical client/server architecture, the three layers described above form natural separations for client and server services. The *business processing logic* layer is typically the *master*, requesting service from the other two layers as the application needs it. The *presentation logic* and *data management logic* layers, in turn, act as *slaves* to the business processing logic layer accessing data, displaying information, requesting the update of a record from a DBMS, and so forth.

Important benefits are provided by this layered architecture:

- *Layer independence*. Each layer is only aware of the services provided by the layer immediately above and below it, but not of the implementation of the lower or higher layers.

- *Flexibility*. An implementation change in one layer (new technology, different hardware, etc.), should not affect the layers above and below it.

- *Simplified implementation and maintenance*. The breakdown of the system functionality into smaller, simpler sections supports this.

- *Standardization*. Standards are developed more easily when the layers of functionality, services, and interfaces are precisely defined in the architecture.

3.4 Application Program Interface (API)

Some newly emerging facilitators make the job of moving to client/server more direct than in the past. Operating systems are becoming more open and have built-in calls or interfaces (APIs) that allow task-to-task communications and resource sharing. These calls are themselves becoming standardized so that applications can be ported from platform to platform with minimal effort.

An Application Program Interface is defined as a programming language interface between a program and its user. The API is implemented by the application platform component performing the service. To promote portability, no vendor extensions should be used. In those instances where a standard API and respective products do not exist, a proprietary interface may be necessary. This interface should be replaced by a standard conforming interface when available. Support applications should be developed to isolate this interface so that replacement impact is minimal.

Sockets

A socket is an endpoint of communication to which a name can be attached. The "type" of socket determines the method and mode of data transfer:

- A *Stream Socket* provides bidirectional, reliable, sequenced, and unduplicated flow of data without record boundaries.
- A *Sequenced Packet Socket* provides bidirectional, reliable, sequenced, and unduplicated flow of data with record boundaries.
- A *Datagram Socket* provides unreliable packet transfer across the socket.
- A *Raw Socket* provides access to underlying communication protocol and is provided so sophisticated Internet applications can be developed.

The client, as well as the server, must first create a socket by issuing some appropriate socket interface call, a bind call, and a connect call. The client and the server can now transfer data by using the "read/write" calls.

1. Berkeley Sockets

Based on the Berkeley Software Distribution (BSD) operating system, network I/O in UNIX environments rely on sockets. Sockets can be perceived as the result of a combination of the port address with the local area network address. Unless a customer has access to the operating system source code, sockets cannot be added.

There may be a large number of sockets on any given TCP/IP network. Each socket uniquely identifies one specific application on a specific node on a specific network. The IP can transmit packets to the appropriate nodes; the TCP then delivers the packets to the appropriate program (process) on that node.

When one socket wishes to connect to another, it must use a specific mechanism to establish that connection. TCP is a connection-oriented protocol that provides these mechanisms. Typically, a connection is established by using an *active* or *passive* network *open*. A socket announces that it is open and available for incoming traffic when it wishes to receive data. This is a passive open. A passive open may be *fully specified*, which means that the socket that issued a passive open tells the network which socket may connect to it.

The active open socket tries to find a socket that it wishes to connect to. The only way an active open can be successful is if the requested socket cooperates by being a passive or an active open. A three-way handshake is used to support correct synchronization between the two end points at connection.

- A synchronization signal and an initial sequence number is sent to the destination by the requestor of the connection.

- The receiver sends back the acknowledgment, sequence number, and synchronization signals when the synchronization signal is received.

- The requestor sends the acknowledgment back to the receiver following receipt of both signals.

Messages can be lost, delayed, duplicated, or delivered out of order since TCP is built upon unreliable packet delivery services (IP). This makes the three-way handshake necessary to ensure proper synchronization. The three-way handshake guarantees that both sides are ready to transfer data, and both have agreed on the initial sequence numbers. TCP uses packet sequencing to ensure proper order of packet delivery and checksum for error detection. The checksum method also augments the guaranteed delivery protocol by using retransmission of messages in the event of time-outs. TCP provides for data transfer from one socket to another. TCP's most used methods for data transfer are as follows:

- Segmented data transfer that allows TCP to send data in segments across the network. To provide for the best efficiency, segment sizes can be adjusted.

- Push mode that forces TCP to send all data without network efficiency considerations being involved.

2. Remote Procedure Calls (RPCS)

RPCs provide a relatively high-level application interface to client/server services that is similar in appearance to normal C language function calls. The need exists for individual process components of an application to run elsewhere on a network. The use of a traditional programming construct, the procedure call, is used by RPCs and extended from a single system to a network of systems. In its communication system role in a client/server environment, the RPC requesting a service from a resource component (server) is issued by a process component (client). The location of the resource component is hidden from the user (client). RPCs provide powerful tools that are necessary to build client/server applications. Two major tools are as follows:

- A language and a compiler produce portable source code simplifies client/server applications development.

- The system architecture and network protocols are provided transparently to the application by a run-time facility that allows distributed applications to run over multiple, heterogeneous nodes.

In order to develop an open and compliant client/server application, a developer creates an interface definition using the *Interface Definition Language (IDL)*. The syntax is similar to ANSI C, but has additional language constructs specifically for a network environment. The IDL compiler then translates the definitions into stubs that are bound with the client and the server as depicted in figure 3-5. The stub on the client system acts as a substitute for the required server procedure. The server stub substitutes for a client in a similar way. Otherwise manual operations, such as copying arguments to and from RPC headers, converting data

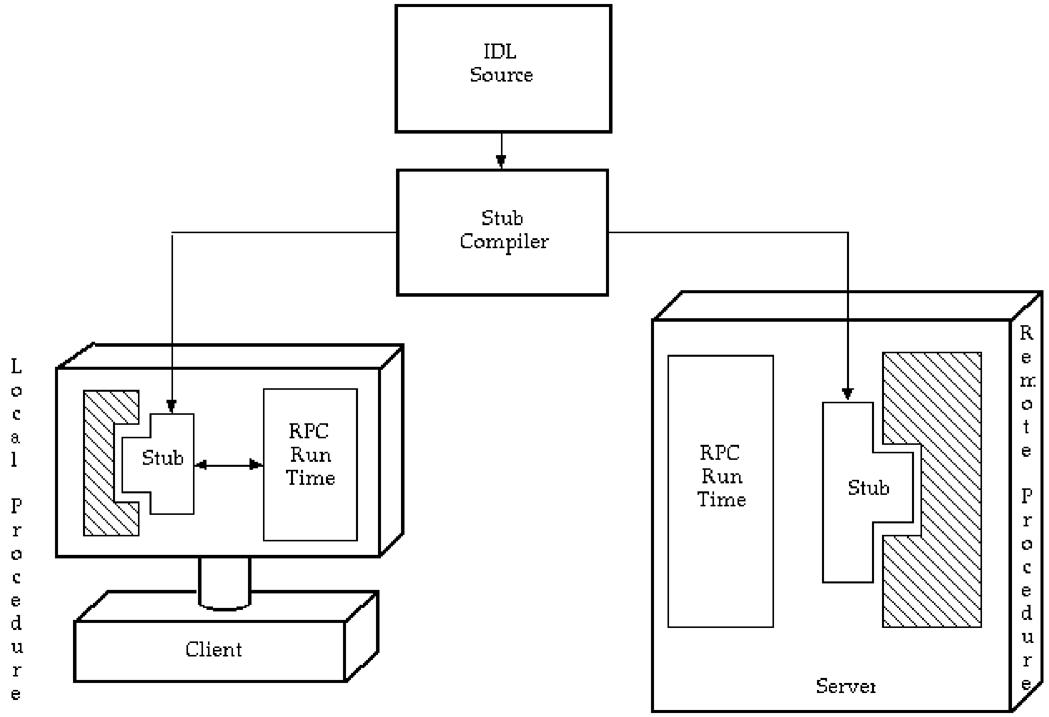

Figure 3-5 RPC implementation

when required, and calling the RPC run- time are automated with the use of stubs. The following features are suggested for RPC run- time:

- Transparency and independence from the underlying networking protocols
- Support for reliable transmission, error detection, and recovery from network failures
- Support for a common method of network naming, addressing, and directory services, which are also independent of network directory services
- Ability to handle multiple requests simultaneously and multi- threading support for parallel and concurrent processing (reducing the time required to complete an application)

- Portability and interoperability with various system environments
- Support for resources integrity and application security

Several methods to implement RPCs exist including those available through Sun's Network Computer Architecture (NCA) and OSF's DCE. They are not, however, interoperable. Once standardized, DCEs RPC may be one of the strongest candidates for a standard RPC.

Server-to-server and client-to-server DBMS connectivity is also an activity best supported by the RPC. By providing a message-based, connectionless mechanism, the remote procedure call allows one process to execute another that resides on a different or remote system, or even running under a different operating system. A database RPC is a request for a service or data issued over a network to a DBMS server by a client or another server. Database RPCs are not like traditional remote procedure calls. Database RPCs can call stored procedures and allow the DBMS server to return multiple records (rows) in response to a single request. This can greatly reduce the network traffic by eliminating the need for a client to send lengthy SQL statements and receive individual records separately. Heterogeneous DBMS connectivity is more easily obtained because language syntax compatibility problems are eliminated. A note of caution is in order, however, about using embedded SQL. Even though the SQL may be ANSI standard, different compilers may not be. Each local DBMS "speaks" its own data access language and the translation of an SQL query into a hierarchical data access, for example, is a major task. Sometimes it cannot be done without rewriting a query or an entire application.

3. High-Level Interface

These communications mechanisms provide an abstraction from the underlying operating system, the network protocols, and network software and hardware. This abstraction is the result of a regulated and carefully designed set of APIs that remain unchanged across various platforms. The interfaces are vendor independent and may not even need to be recompiled when porting from one platform to another.

A. Hiding Lower-Level API

OPENLINK is a software suite that provides a general-purpose client/server API. This allows a client application to be developed without the need for the application programmer to understand or develop the sockets protocols necessary to connect to and communicate with a server (see Section 4.5, Other Tools).

B. Language Bindings Required

Binding refers to the binding of a conceptual set of services and a standardized interface that establishes rules and syntax for accessing them. Language bindings are documents specifying POSIX in programming languages.

Working groups are reformulating the POSIX 1003.1 standard to include such programming languages as C, Ada, and FORTRAN-77. The core services of this standard are programming language-independent and represent services that are common to the languages. Some fundamental system services cannot be included in the core section. For example, though memory management may at first be considered a core service, it must be included in the language-specific section of the standard. Programming languages such as FORTRAN have not typically provided memory management, and categorizing memory management as a core service would impose unreasonable requirements for FORTRAN implementations.

It is not unreasonable for other language bindings to specify some areas that are undefined or unspecified by the underlying language standard, or that are permissible as extensions. This may solve some difficult problems.

Using as much as possible of the target language in the binding strengthens portability. If a program wishes to use some POSIX capabilities, and these are bound to the language statements rather than appearing as additional procedure or function calls, and the program conforms to the language standard while using those functions, it will port to a greater range of systems than one that is required to use procedure or function calls instituted specifically for the binding to POSIX to do the same thing.

A program that requires the POSIX capabilities that are not bound to the standard language directly (as discussed above) has no potential for portability outside the POSIX environment. It does not matter whether the extension is syntactic or a new function; it still will not port without effort. Given this, it seems unreasonable not to consider language extensions when determining how best to map the functionality of POSIX into a particular language binding.

Binding directly to the language, where possible, should be encouraged both by making maximal use of the mapping between the operating system and the language that naturally exists, and where appropriate, for the languages to request changes to the operating system to facilitate a better mapping.

Chapter 4

Tools

4.1 Introduction

Various tools are required in the support of development, implementation, and deployment of client/server applications. This chapter outlines those tools that are required, those without which client/server applications cannot be deployed, and those that make the job of client/server development easier.

4.2 IBM-Compatible Mainframe Tools

A high percentage of computer systems that manage DoD corporate data and that are necessary for emerging applications are mainframes. Most are IBM S/370 compatibles running the proprietary IBM MVS operating system; most of these in turn are MVS/XA. The natural communications protocols in this environment are not the open TCP/IP protocols necessary for client/server processing as described in this paper, but rather proprietary SNA protocols requiring SNA communications front-end processors.

In order to implement servers in the mainframe hosts to supply data on demand to clients using open protocols, and perhaps using some application processing as well, these hosts must be augmented with TCP/IP components. This augmentation includes software, the TCP/IP drivers through which applications communicate with the underlying protocols, and hardware for connection to a LAN local to the host's IPC.

1. TCP/IP LAN Adapter

Several viable sources of TCP/IP Ethernet LAN adapters are available for IBM-compatible mainframe systems. When considering the selection of a TCP/IP network adapter and software for the mainframe, performance of the subsystem, and its impacts upon the performance of the mainframe computer system must also be considered. Some of the hardware/software suites support the off-loading of TCP/IP protocol processing to the controller.

The following list of mainframe TCP/IP LAN adapters is not all-inclusive; other products may exist. These products were known at the time this paper was written:

- InterLink 3722
- IBM 3172
- Fibronics K2000

2. TCP/IP Software

Software for IBM-compatible mainframes is generally available from the same sources as for the LAN adapter hardware. All of the following products have NFS software available. The software packages corresponding to the adapters listed above are as follows:

- InterLink SNS/TCP/IP
- IBM TCP/IP for MVS, X Window, and Kerberos is available for the IBM suite.
- Fibronics KNET

3. TCP/IP Sockets Support Library

All three vendors noted above have a TCP/IP sockets library available as an option. The sockets library is required for the development Central Design Activity (CDA) installations, with code linked into the applications.

4. C Compiler

A C compiler is required for the development CDA in order to create COBOL or Ada language bindings to the sockets functions. A C compiler is also required for the mainframe port of the OPENLINK product, as discussed below. At least two viable C compiler products are available for the mainframe:

- IBM C
- SAS C

The TCP/IP sockets library vendor should be consulted to ensure that any compiler dependencies are addressed and that the correct compiler is purchased.

A run-time library is available for both the IBM and SAS compilers. Using the run-time library may reduce the size of load modules, but may lead to licensing issues. The run-time library, if used, must be present in each mainframe upon which the load module runs. The SAS compiler will create load modules that do not require a run-time library with the overhead of a larger load module.

4.3 UNIX/POSIX Minicomputer Tools

All UNIX/POSIX implementations available commercially and on contract vehicles have the necessary hardware and software available for client/server implementations. Most of these components are optional, however, and must be specifically ordered. It is recommended that *all* UNIX/POSIX minicomputers, whether for the department or desktop, be ordered with the following hardware and software components:

- TCP/IP LAN adapter
- TCP/IP software
- OSI gateway software
- NFS software
- X Window client software

All departmental UNIX/POSIX systems ordered for application development should include the following:

- TCP/IP sockets support library
- C compiler
- Ada compiler
- X Window development tools, including a GUI builder, and an Ada language binding library

4.4 MS-DOS/Windows PC Workstation Tools

The tools required for MS-DOS/MS-Windows workstations are available from contract vehicles available to the DoD. The basic components required include the following:

- TCP/IP LAN adapter card
- TCP/IP driver software
- NFS software
- X Window server software

For those systems used by application development programmers, the following are also required:

- TCP/IP sockets support library
- C compiler
- Ada compiler

4.5 Other Tools

4.5.1 OPENLINK High-Level Application Program Interface

OPENLINK is a software suite that provides a general-purpose client/server API. This allows a client application to be developed without the need for the application programmer to understand or develop the sockets protocols necessary to connect to and communicate with a server.

The OPENLINK client software is called from an application by a single function call. Data may be passed to the server from the client and received by the client from the server with the single call. The OPENLINK client software is a very small (approximately 5K byte) subroutine that is linked into the client application.

A single OPENLINK server process in each server machine provides service to all OPENLINK clients connecting to the server. The OPENLINK server connects with server application code to process a single service request and returns the resultant data to the appropriate client.

OPENLINK has been ported to a number of platforms, including IBM-compatible mainframes, MS-DOS PCs, and a number of UNIX systems. An OPENLINK server is available on each of these platforms, as is an OPENLINK client.

OPENLINK was developed by the DLA System Automation Center (DSAC) in Columbus, Ohio. Further information may be obtained from DSAC. Points of contact are Mr. Carl Litzinger, DSAC-FS (614) 692-8107 or Mr. Steve Hinchman, DSAC-O, (614) 692-9646.

4.5.2 Online Report System (ORS)

ORS is not a client/server process, but provides a means for distributing and viewing paperless reports generated by applications that are remote to the end user. It is included in this document to publicize the need for and a solution to the necessity for processing, distributing, and viewing report data on a workstation in a logical and easily understood fashion.

Nearly all business data systems provide printed reports that are used by functional users for many tasks. These reports are sometimes very large, difficult to manage in a relatively centralized workplace, and nearly impossible to manage if they need to be distributed to many sites around the country remote from the data processing installation (DPI) that generated them. As DPI consolidation progresses, the problem is expected to worsen. With time, data system redesign will likely eliminate the need for printed reports, replacing them with on-line information. However, during the interim, the need for printed reports will continue.

ORS allows reports to be "printed" to files, transmitted to a "report server" minicomputer at the end users' site in electronic form, and viewed by one or more end users at the site. ORS is parameter driven so that it can be set up for a user to view only that report information in which he or she is interested or is authorized to see. Standard UNIX user and group permissions are used to provide report privacy and user authorities. ORS removes heading information from the reports, except for a single occurrence that stays on the screen. The report may then be scrolled horizontally as well as vertically. Text search capabilities are provided.

ORS was developed by the DLA System Automation Center (DSAC) in Columbus, Ohio. Further information on ORS is available from DSAC-T. Point of contact is Mr. Douglas Whipple, DSAC-TOS, (614) 692-9070.

Chapter 5

Economic Analysis

5.1 Economic Analysis Factors

While technology such as client/server processing can be an enabler for vastly improved applications, access to information, end-user satisfaction, and much higher flexibility in the deployment of applications, the decisions to implement client/server technology must be based on functional requirements. Client/server technology will, in many cases, require the implementation of new technical infrastructure such as LANs, workstations suitable for cross-functional use, and data and application servers. These infrastructure changes will be costly and in some cases will impact the working environment as well as the technical infrastructure itself. These costs must be borne by the functional community for which the infrastructure changes are being undertaken. Savings in the functional processes will have to justify the costs of the new technology.

We in the technical community cannot provide functional cost savings that will justify new technologies such as client/server. We can, however, provide insights into the costs of implementation of new technology, and assist in bringing these technology costs into the Functional Economic Analysis equation. The technical community can also assist the functional community in determining whether and how new technology such as client/server can form the basis for implementing business process changes that might not be possible without new technology.

5.2 New Technology Cost Categories

1. Technology Infusion

The cost of the new technology to be installed is a direct cost to the implementation of client/server applications. This category includes the following:

- Hardware: Workstations, LAN components, servers, and routers
- COTS Software: Workstation and server software, including the cost of ongoing support of this software
- GFE Software: Utility and system support software, as well as the cost of ongoing support. This would include public domain or "freeware" products and their support.

2. Training

- System Administration: LAN, server, and workstation support staff that directly supports the infrastructure on a daily basis.

- Applications programmer: Client/server development training and conference costs.

- CDA System Programmer: LAN, server, and workstation system software training required to develop these highly skilled technical software positions.

- End User: Functional end-user training in the interaction with the GUI, mouse, and other workstation or server features, as well as training for the newly implemented applications. Consistency in the implementation of the GUI between applications and across functional areas will reduce this cost.

3. LAN Administration

- System Administration: The ongoing cost of the technical staff that provides LAN, server, and workstation support directly to the infrastructure at a functional user's site.

4. Environment

- Work Area Expansion: This would include changes to the user's work area, such as size expansion to accommodate a desktop computer where no computer or terminal existed.

- Electrical Power Enhancement: This would include the costs necessary to update electrical power in a work area when the power previously did not have to be adequate to support desktop computers.

APPENDIX TABLE OF CONTENTS

Exhibits

Executive Summary

The DISA logistics Technical Integration Management office currently has over 20 near-term initiatives (NIIs) in the process of becoming baseline systems. In general, these systems will require new hardware acquisitions to support implementation of the baseline system and subsequent migration to DoD standard systems. The baseline systems can consist of hardware deployed in three tiers: centralized systems (DSOFs), multi-user distributed systems, and single-user distributed systems. This document addresses the identification of potential contract vehicles to be used to purchase hardware for each tier.

The centralized systems support an IBM/MVS-compatible execution environment with wide area communications services provided via the Defense Information Systems Network (DISN). Both the multi-user and single-user distributed systems must support a Portable Open Systems Interface Standard (POSIX) execution environment for newly acquired hardware.

Hardware requirements at the centralized tier therefore consist of IBM compatible mainframes, peripherals, and communications equipment. A number of different potential contracts were reviewed to determine if they could be used to purchase the required ADP hardware. A complete list of all contracts reviewed is presented in the body of this document. Those contract vehicles that appear most applicable to supporting DISA/TIM logistics centralized system hardware acquisition requirements are identified in exhibit ES-1. Both existing contract vehicles and contracts that can or will be awarded in the next 12 months have been included. No single existing contract vehicle will meet DISA/TIM logistics requirements for both IBM compatible mainframes and peripherals. However, two acquisitions in process that may help with CPU requirements: the GSA Drop-In Technology acquisition and DLA CPU acquisition. Both support purchase of IBM compatible mainframe equipment. The DLA contract will be awarded in late 1992 and the GSA contract in late 1993. Neither includes DASD or tape systems. Disk and tape subsystems may be available from the DLA contract with Storage Technology but will likely require a separate delegation of procurement authority (DPA) from GSA. If a separate DPA from GSA is required, GSA may insist on full and open competition.

The DLA CPU contract can be used to purchase IBM compatible mainframes based on the scope of the contract. Major acquisitions using this vehicle for non-DLA sites will, however, require a separate DPA from GSA. GSA will likely require a separate competitive acquisition to obtain the DPA. GSA's position could make this vehicle of little value to other than DLA sites.

An alternative option for purchasing IBM compatible mainframes and peripherals is to use Small Business Administration (SBA) 8(a) businesses that qualify as regular dealers. It appears that these acquisitions can be completed in less than 4 to 5 months and can be structured as Indefinite Quantity/Indefinite Delivery (ID/IQ) contract vehicles supporting the requirements of more than one near-term initiative. The primary advantage of this approach is the ability to structure the contract to meet specific NII requirements without being required to conform to the SBA 50 percent services content requirements of other 8(a) contracts such as the DISA Omnibus contract with Advance Corporation. For those hardware components that cannot be purchased through existing contracts, using this approach is recommended and discussed in more detail in the body of the document.

Multi-user and single-user distributed system hardware requirements include UNIX servers, local area network hardware components, X Terminals, personal computers (UNIX), and DISN-to-Ethernet gateways. Recommended contract vehicles for acquisition of these components are provided in exhibit ES-2. These contract vehicles appear to easily meet near-term logistics requirements. The most probable contract vehicles are AFCAC 300 and Desktop IV. While these components can be purchased through existing or soon-to-be-awarded contracts, the overall quantities required to support longer term logistics requirements will probably not be available on these contract vehicles. A separate acquisition to meet logistics requirements for UNIX servers, workstations, X Terminals, and LAN components would be advisable. It may also be possible to obtain these components through in-process acquisitions if they can be structured to include logistics requirements. One candidate acquisition for including longer term logistics requirements is the Standard Engineering Workstation (SEWS) acquisition currently in process by the Air Force at Wright Patterson AFB. It is difficult to use SBA 8(a) contract vehicles for these requirements because of the SBA 50 percent services content rule for 8(a) contracts. The 50 percent content exception discussed earlier applies only to mainframe purchases through 8(a) contracts.

The contract vehicles discussed in this document appear to provide the means of purchasing small quantities of ADP hardware for the NIIs but will not support larger quantities of workstation and mainframe equipment potentially required in the more distant future (more than 1 year out). The DISA/TIIM office should work with DISA/CIM to establish longer term purple contracts for required ADPE for both the centralized systems and user access area (distributed systems) environments. Failure to take this action now will lead to serious acquisition problems as NIIs are deployed.

Exhibit ES-1, Centralized System Acquisition Vehicles

Contract	Gov./Vendor Contact	Summary
DISA Omnibus Contract Advance Corp. Melvin Laney (202) 785-9220	John Godbey - CO (703) 692-2782 Rhonda LaGard - Contract Specialist (703) 692-3706 Diana Hickey (703) 692-9685	DISA 8(a) contract for IBM compatible minframes, peripherals, and communications equipment. Requires 51 percent services content for each content for each order.
GSA Drop-In Technology	Ann Liddle (703) 756-4500	Acquisition in process for IBM compatible mainframe equipment. Draft RFP in late October 1992 with award expected in late 1993. No peripherals or communications equipment included.
AIPC	Army IPCs Gloria McGee (703) 325-3334	Potential vehicle for Amdahl peripherals including DASD, channel adapters, and cache memory. Purchasing limited to Army IPCs. Other contracts address tape libraries, cartridge tape systems, and upgrades to mainframe CPUs. All equipment must be installed at Army IPCs.
DLA Consolidation	Chuck Wagner (703) 274-5352	Supports acquisition of IBM compatible mainframe equipment for DLA. Scope supports all DoD but a separate DPA will be required for non-DLA requirements. Contract award expected in late 1992.

Exhibit ES-1, Centralized System Acquisition Vehicles (cont,d)

Contract	Gov./Vendor Contact	Summary
		Does not include DASD or tape systems.
DLA/Storage Tech.	Phil Silas (703) 274-3210	DASD and tape library add-on peripherals for DLA sites. Potentially usable for non-DLA requirements.
Marine Corps CPU Augmentation	Kenny Bluteau - COR (703) 696-1010	Augment existing Amdahl installations within Marine Corps.

Exhibit ES-2, Distributed System Acquisition Vehicles

	Gov./Vendor	
Contract	Contact	Summary
Small Multiuser Computer (SMC)	Mary Morris DLA-ZIA Bldg. 3, Cameron Station (703) 274-7506	Usable for 386 PC hardware, software, and PC LAN equipment.
Desktop IV	Maj.J.D. Smith Gunter AFB (205) 416-3464	Usable for 486 PCs, software, and PC LANs. Award currently being
protested.		
DECCO-BOAs	DECCO Scott AFB (618) 256-9690	Usable for purchase of PC hardware only at present. No OS or application software included in standard configurations.
SEWS	Barry Cain - CO (513) 257-6721 Chuck Schneggenburger (513) 255-3366	Acquisition in process for engineering workstations. DISA could request that NTI requirements be added to acquisition.
GTSI GSA Sched	GTSI Sales (800) 999-4874	PC hardware and software, communications equipment, LANs, UNIX workstations, Terminals GSA GKOOK92AGS6084.
GMR GSA Sched	GMR Sales Rep. (800) 232-4671	PCs, LAN equipment, PC software, communications equipment, printers, etc. GSA GSOOK92AGS5451.
Sylvest GSA Sched	Maribeth Girard (301) 459-2700	HP workstations, Sun workstations, and Tektronix X Terminals included in schedule.

Exhibit ES-2, Distributed System Acquisition Vehicles (cont'd)

Contract	Gov./Vendor Contact	Summary
PG-LAN	Jakki Rightmeyer Shirley Dumbar NCTAMS LANT Code N811.2 138 E. Little Creek Norfolk, VA 23505 (804) 445-1493	Usable for Ethernet LAN equipment and DEC PCs
ULANA EDS	Dusty Wince (703) 742-2406	LAN equipment for both Ethernet and FDDI LANs. Limited to $25K purchase. This vehicle is a BOA.
ULANA II	Lt. King Tinker AFB (405) 734-9773 WendelCrittenden (405) 734-9928	LAN equipment including DISN gateways, FDDI and Ethernet LANs, and MVS gateways. This acquisition is currently in process with award expected late 1993.
AFCAC 300	Capt. Brown - AFCAC (617) 377-8638	UNIX servers, terminals, UNIX workstations, LAN equipment. Still under acquisition. Award expected in mid-October 1992.
AFCAC 305	Acquisition in process	UNIX-based database machines. Can be used to provide UNIX servers for DISA/TIMXITIs.

Appendix A
Government Contractual Vehicles Supporting Client/Server Development

The following contract vehicle list was developed for the Technical Integration Management Office for Logistics, JLSG-TI. This list includes a general description of products offered on the contract, and contacts for each contract who can provide detailed information.

Of the contracts listed, one of the most promising for providing client/server components on the personal workstation and server tiers is the Navy Super-mini (AFCAC 300) contract. At the time of this writing, acceptance testing was under way and contract details could not be released. We will continue to monitor this contract closely and will provide more complete details as they become available.

One of the areas where client/server components are needed most is in conjunction with the IBM compatible mainframe. The only DoD contract that provided such components was the ULANA contract, which has expired. To our knowledge, no current contract vehicle exists outside the GSA schedule contract from which IBM mainframe TCP/IP components can be purchased.

Summary of Contracts Applicable
to
DISA/TIM Logistics Acquisition Requirements
for
Near-Term Initiatives

prepared for

Defense Information Systems Agency
Technical Integration Management Office
Logistics

Wright Patterson AFB, Ohio

30 September 1992

prepared by

Data Networks Corporation
1738 Dressage Drive
Reston, Virginia 22090
(703) 438-8499

Section 1 - Introduction and Background

Logistics community Central Design Activities (CDAs) have both near-term and long- term requirements to acquire ADP equipment to met Corporate Information Management (CIM) requirements. These requirements include equipment needed to support both near-term central-site consolidation efforts and the installation of single- and multi-user distributed systems. In addition, the DISA/TIM office for logistics has near-term requirements to purchase small quantities of ADP equipment to validate technology applicable to the near-term initiatives and to validate the NII application operation prior to deployment at the DSOFs. This document identifies potential acquisition approaches and existing contract vehicles that can be used to acquire the ADP equipment necessary to meet NII objectives over the next year.

The ADP equipment required to meet NII requirements includes IBM MVS-compatible mainframes, peripherals, and communications equipment; UNIX workstations and servers; X Terminals, GOSIP local area networks, personal computers with UNIX and X Window, and DISN/LAN gateways.

In general, the existing ADP support in the logistics community consists of IBM 3270-

compatible display stations accessing IBM/MVS-compatible mainframes. The communications environment is bisynchronous and SNA through DISN. In some cases, asynchronous device access to IBM compatible mainframes is supported through LANs and gateways to the IBM environment. User access area equipment must be replaced or augmented to support access to the proposed CIM architecture standardizing on POSIX and GOSIP, with X Window and SQL providing standard GUIs and database access. Rather than using bisync and SNA as the primary communications protocols, the CIM architecture requires GOSIP protocols to support user access area to application processor interfaces.

A complete list of contract vehicles investigated and a summary of whether or not they can be used to support NII requirements are provided in exhibits A-1 through A-3. Exhibit A-1 summarizes contract vehicles that support acquisition of IBM compatible mainframes, peripherals, and communications equipment. Exhibit A-2 addresses personal computers and workstations and exhibit A-3 addresses communications equipment and local area networks. Detailed data for each contract is provided in the last three sections of this appendix.

Exhibit A-1. Contract Vehicles for IBM Mainframes and Peripherals

Contract	Gov./ Vendor Contact	Summary
DISA Omnibus Contract Advance Corp. Melvin Laney (202)785- 9220	John Godbey - CO (703) 692- 2782 Rhonda LaGard - Contract Specialist (703) 692- 3706 Diana Hickey (703) 692- 9685	DISA 8(a) contract for IBM compatible mainframes, peripherals, and communications equipment. Requires 51 percent services content for each order.
GSA Drop- In Technology	Ann Liddle (703) 756- 4500	Acquisition in process for IBM compatible mainframe equipment. Draft RFP in late October 1992 with award expected in late 1993. No peripherals or communications equipment included.
AIPC	Army IPCs Gloria McGee (703) 325- 3334	Potential vehicle for Amdahl peripherals including DASD, channel adapters, and cache memory. Purchasing limited to Army IPCs. Other contracts address tape libraries,
cartridge		tape systems, and upgrades to mainframe CPUs. All equipment must be installed
at		Army IPCs.

Exhibit A-1. Contract Vehicles for IBM Mainframes and Peripherals (con't)

Contract	Gov./Vendor Contact	Summary
GMR 8(a)Reseller	GMR Sales Rep. Chuck Yarborough (703) 263-9146	IBM compatible mainframes, peripherals, and communications equipment via 8(a) vehicle. SIC 5045.
DC Information Systems	DCIS Sales Rep. Wayne Carter (301) 585-6103	IBM compatible mainframes, peripherals, and communications equipment via 8(a) vehicle. SIC 5045.
TSS, Inc.	Charita Albright - SBA (202) 634-1500 x307	IBM compatible mainframes, peripherals, and communications equipment via 8(a) vehicle using 5045 exception to content requirements.
U.S. Marine Corps CPU augmentation	Kenny R.Bluteau - COR (703) 696-1011	Augments existing Amdahl installations within Marine Corps.
Marine Corps Front-End	Cathy Ziegler - CO (703) 696-1010 (MCFEP)	Supports acquisition of Processor communications controllers, token ring interfaces, Ethernet interfaces, and maintenance for the Marine Corps.
DLA/Storage Tech.	Phil Silas (703) 274-3210	DASD and tape library add-on peripherals for DLA sites.

Exhibit A-1. Contract Vehicles for IBM Mainframes and Peripherals (con't)

Contract	Gov./Vendor Contact	Summary
DLA CPU Purchase	Chuck Wagner (703) 274-5352	IBM compatible mainframes are being acquired through this vehicle. Purchases are targeted to DLA-related activities. Contract award is expected in late 1992. Other DoD components may purchase from this contract but a separate DPA is required per discussion with C. Wagner.

Exhibit A-2. Personal Computer and Workstation Contract Vehicles

Contract	Gov./Vendor Contact	Summary
Small Multiuser Computer (SMC)	Mary Morris DLA/ZIA Bldg. 3 Cameron Station (703) 274-7506	Usable for 386 PC hardware, software, and PC LAN equipment.
Desktop IV	Maj.J.D. Smith Gunter AFB (205) 416-3464	Usable for 486 PCs, software, and PC LANs. Award currently being protested.
TAC III	Mary AnnGroomes (202) 433-2387	Not usable for logistics requirements, limited by Warner amendment to tactical requirements.
RCAS	Mr. Tompkins (703) 339-1741	Contract addresses turnkey system delivery, not ADP hardware. Therefore not usable for CIM NTI requirements.
NPIC	Grant Chisolm (202) 863-3750	It is highly probable that this contract will support some UNIX workstation acquisition requirements. More information should be available after 1 October. The contract is currently undergoing renewal negotiations.
AFCAC 308	Mrs. Durette (804) 764-4503	Acquisitions limited to specific AF programs due to lack of available quantities.

Exhibit A-2. Personal Computer and Workstation Contract Vehicles (cont'd)

Contract	Gov./Vendor Contact	Summary
JCALS	Wayne Johnson- CO (908) 532-0416 Jim Tomaldson - PM (908) 532-0400	Not usable. This contract is not written to support equipment-only purchases.
DECCO BOAs	DECCO (618) 256-9690	At present, usable for Scott AFBpurchase of PC hardware only. No OS or application software included in standard configurations.
SEWS	Barry Cain - CO (513) 257-6721 Chuck Schneggenburger (513) 255-3355	Acquisition in process for engineering workstations. DISA could potentially request that N TI requirements be added to acquisition.
GTSI GSA Sched	GTSI Sales	PC hardware and software, (800) 999-4874 (801) communications equipment, LANs, UNIX workstations, X Terminals. GSA GKOOK92AGS6084
GMR GSA Sched	GMR Sales Rep.	PCs, LAN equipment, PC (800) 232-4671 software, communications equipment, printers, etc. GSA GSOOK92AGS5451
HFSI GSA Sched	Elizabeth Fox (703) 827-3160	Add-In SPARC cards for PCs.
Sylvest GSA Sched	Maribeth Girard (301) 459-2700	HP workstations, Sun workstations, and Tektronix X Terminals included in schedule.

Exhibit A-2. Personal Computer and Workstation Contract Vehicles (cont'd)

Contract	Gov./Vendor Contact	Summary
Netcap	Maj.Schaeffer (617) 271-8848	Tektronix X Terminals. Not usable for NTI requirements. Scope limited to IDHS and DODIIS users.
AFCAC 300	Capt. Brown - AFCAC (617) 377-8638	UNIX servers, X Terminals, UNIX workstations, and LAN equipment. Still under acquisition. Award expected in mid-October.
AFCAC 305	Under acquisition	Database machines based on UNIX servers.
WIS Workstation	Joseph P. Cloutier - PM,Gunter AFB (205) 416-5768 Bud Smith - PM HFS Inc.	Tempest and non-Tempest Macintoshworkstations. Workstations running AIX (UNIX) OS. Scope includes all departments for command and control applications. Probably not available for logistics applications.

Exhibit A-3. Communications Equipment Contract Vehicles

Contract	Gov./Vendor Contact	Summary
PG-LAN	Jakki Rightmeyer Shirley Dumbar NCTAMS LANT Code N811.2 138 E. Little Creek Norfolk, VA 23505 (804) 445-1493	Usable for Ethernet LAN equipment and DEC PCs.
ULANA EDS	Dusty Wince (703) 742-2406	LAN equipment for both Ethernet and FDDI LANs. Limited to $25K purchase. This vehicle is a BOA.
ULANA II	Lt. King Tinker AFB (405) 734-9773 WendelCrittenden (405) 734-9928	LAN equipment including DISN gateways, FDDI and Ethernet LANs, and MVS gateways. This acquisition is currently in process with award expected in late 1993.
AFCAC 300	Capt. Brown- AFCAC (617) 377-8638	UNIX servers, X Terminals, UNIX workstations, and LAN equipment. Still under acquisition. Award expected mid-October 1992.

Section 2 - Legacy System Environment

2.1 Introduction

In the legacy system environment, IBM or IBM compatible mainframe processors, memory, peripherals (especially data storage devices), and communications equipment are potentially required. In addition, software to support the additional equipment or to support required interfaces may be required. Two methodologies for acquisition of add-on equipment and software for the legacy system environment were identified.

The first method involves using existing ID/IQ contract vehicles held by the various services and commands or the GSA schedule contracts for IBM and IBM compatible mainframe vendors. A 2-week search for such contract vehicles through contacts in the mainframe vendor community has yielded no contracts that would be usable by more than one of the CDAs. While further searching may lead to a contract vehicle or ongoing acquisition that better meets DISA near-term and migration system requirements, it does not appear likely. All contracts reviewed for potential use as vehicles to purchase legacy system hardware and system software are listed in exhibit A-1.

At present, SBA 8(a) contracts appear to be the most useful acquisition vehicles. An August 1991 SBA ruling granted a waiver to 8(a) contractors holding the Standard Industrial Classification Code 5045, and otherwise qualified as regular dealers under the Walsh-Healey Act, to sell mainframe equipment to the Federal Government without the 51 percent added-value requirement normally imposed on hardware purchases under the 8(a) set-aside program. Our initial assessment is that this method may allow the DISA TIIM office to develop an open, timely, and accessible vehicle for acquisition of legacy system environment equipment.

2.2 Mainframes and Peripherals

2.2.1 Existing Contract Vehicles/GSA Schedules

Other than GSA schedules, our initial search for ID/IQ contract vehicles yielded few possibilities. The three potential suppliers of this equipment are International Business Machines (IBM) Corporation, Amdahl Corporation, and Hitachi/NAS, in addition to other companies that manufacture peripheral and communications equipment. Of these suppliers of mainframe hardware, none has a contract vehicle to sell to all DISA activities except through an 8(a) program contract requiring 51 percent services content (DISA Omnibus contract with Advance Corporation) of GSA schedule. IBM has no existing ID/IQ contracts usable by DISA or DoD that we could find. Amdahl has contracts to sell to the Army IPCs and the Marine Corps, and Hitachi has no contracts usable by DoD. In fact, no contract vehicle other than a very limited GSA schedule exists for Hitachi/NAS mainframe equipment. This GSA schedule includes primarily maintenance and add-on features to existing Hitachi and NAS mainframes. See the Mainframe Contract Vehicles Descriptions later in this appendix.

For Amdahl equipment, one existing contract for central system equipment, one for communications controllers, and one for DASD were identified. For all three contracts, however, accessibility was limited to the units holding the contracts. For the DASD equipment, the contract is held by the Army IPCs. The mainframe and communications controller contracts are held by the Marine Corps. Overviews for these two contract vehicles are included in the Mainframe Contract Vehicles Descriptions section. Amdahl's GSA schedule for FY 1993 was signed early this month and is currently at press. The company did not have a GSA schedule last year.

No data on existing contract vehicles for IBM mainframes has been identified thus far. An IBM GSA schedule exists and a copy has been requested, though not received. Incremental equipment and software can no doubt be obtained in this manner.

Two different acquisitions are in process that could potentially be useful. The first is the GSA Drop-In Technology acquisition that addresses purchase of IBM compatible mainframe equipment by Government agencies and departments. The second is the DLA Consolidation acquisition. Both are expected to be awarded in late 1993. Both acquisitions are summarized in the Mainframe Contract Vehicles Description section.

2.2.2 8(a) Mainframe Waiver Contracting

In August 1991, the SBA established a waiver of the nonmanufacturer rule for mainframe computers and associated peripheral equipment. The effect of this waiver is to allow an otherwise qualified small business regular dealer to supply such products on a Federal contract set-aside for small business, or awarded through the 8(a) program, without the 51 percent value-added or services requirement imposed when other equipment is supplied in an 8(a) set-aside contract. As implemented, the waiver imposes two basic conditions on the contractor and the Government agency wanting to use this contracting method:

- The 8(a) contractor must have the Standard Industrial Classification (SIC) Code of 5045 (Product and Service Code 7021) in its SBA business plan suite.

- The procuring agency must determine that the contractor is a regular dealer in accordance with the Walsh-Healey Act.

Nationwide, less than ten 8(a) certified companies have the required Standard Industrial Classification Code. Three of these companies have been certified by IBM on their FSC 8(a) protege program:

- DC Information Systems
  8121 Georgia Avenue, Suite 600
  Silver Spring, MD 20910
  Contact: Wayne Carter
  President
  (301) 585-6103

- Government Micro Resources
  14121 Parke Long Court, Unit 104
  Chantilly, VA 22021
  Contact: Chuck Yarborough
  5045 Program Director
  (703) 263-9149

- Technical Software Services

1400 Mercantile Lane, Suite 200
Landover, MD 20785
Contact: Marcus Price
President
(301) 772- 0101

Overviews for these companies are included in the Mainframe Contract Vehicles Descriptions section. A complete list of the 8(a) certified companies nationwide holding the required Standard Industrial Classification Code can be obtained by writing SBA at the following address:

U.S. Small Business Administration, Attn. Janice Harris
Eighth Floor
409 Third Street SW
Washington, DC 20416
Tel: (202) 205- 6410
Fax: (202) 205- 7549

In addition, the U.S. Department of Labor is being asked to identify the key requirements for compliance with the Walsh- Healey Act. Since the contracting agency is required to make the determination for Walsh- Healey Act compliance, audit requirements affect how expeditiously a contract vehicle could be established. Requirements for compliance with the Walsh- Healey Public Contracts Act are as follows:

- Existence of a suitable inventory of the equipment, software, and supplies being sold
- Warehousing space as required by the levels of inventory maintained
- Minimum wages
- Overtime pay requirements
- Child labor prohibitions
- Recordkeeping requirements
- General health and safety standards

SBA annually audits its 8(a) program participants with the 5045 SIC code regarding warehousing and inventory provisions, but does not evaluate on any other requirements.

In the absence of available or accessible ID/IQ contracting vehicles (especially for IBM mainframes), this acquisition method appears to provide a flexible and timely way to purchase add- on equipment and software in the legacy system environment. The 8(a) contractors' representatives are quoting 30 to 120- day elapsed time from the completion of a Statement of Work that includes a potential list of equipment and software to be supplied, to the award of a sole- source ID/IQ contract for mainframe equipment and associated peripherals and communications equipment. The following steps are required:

- The contracting agency and the contractor develop and complete the Statement of Work

- IRM submits the SOW to the cognizant CO with the recommended 8(a) contractor

- CO develops offer letter and sends it to SBA

- SBA reviews SOW and sends acceptance letter to CO

- CO forwards SOW to the selected 8(a) contractor
- Contractor develops technical and cost proposals and submits to CO
- CO and COTR (IRM) review proposal and request clarifications and modifications as required
- Contractor resubmits modified technical and cost proposals
- Price negotiations are conducted
- Contract is forwarded to SBA
- SBA provides final approval

These steps presuppose that the contractor has been certified compliant with the Walsh-Healey Act and that the agency has the appropriate DPA.

Were the ID/IQ contract to be competed among qualified 8(a) companies, the elapsed time from SOW completion to award may take up to twice the quoted time for sole-source contracts. Competing an 8(a) ID/IQ contract would be advisable if DISA TIM wishes to establish a single open ID/IQ contract vehicle using this method. An additional complication is that the ID/IQ SOW must specify all equipment and software that could potentially be required by all the CDAs, especially those engaged in near-term initiatives. However, the time required for the development of a comprehensive list and for the competition may delay a number of near-term initiatives.

An alternative is for each of the CDAs to open its own sole-source ID/IQ contract with an 8(a) 5045 contractor for the equipment and software the CDA would require. The conditions under which this would be recommended are as follows:

- When the CDA has completed a comprehensive engineering process and can identify the array of equipment it will need for its near-term initiative(s)
- When the equipment required includes a mainframe processor (This requirement is stated in the SBA regulation but the 8(a) contractors have stated that waivers can be obtained for this requirement.)
- When the dollar value of the incremental equipment and software justifies this administrative undertaking (The 8(a) contractors contacted were quoting desired guaranteed minimums ranging from $600K to $3M. However, this requirement can probably be negotiated.)

It should be mentioned that Government Micro Resources is currently discussing a Statement of Work with DLA Contracts for an ID/IQ contract that includes IBM compatible mainframes and peripherals. The status of these discussions and the timing of this acquisition will be verified with DLA contracts management.

2.2.3 Mainframe Acquisition Conclusions/Recommendations

The survey for contract vehicles to acquire mainframe and peripheral equipment in the legacy system environment has not yielded a contract vehicle that could be used by multiple CDAs. The 8(a) contracting using the mainframe waiver should be considered by the CDAs with significant equipment needs who have completed their engineering baseline. For the initiatives with limited requirements, only the GSA schedule appears to be available aside from the contracts of limited scope or high service content requirements, as described later in this appendix.

2.3 Peripherals

Two options were explored for acquisition of IBM compatible peripheral equipment. The first consists of identifying existing contract vehicles. The second is to make use of sole-source 8(a) contracts.

No existing contract vehicle supports the acquisition of IBM compatible peripheral equipment for all CDAs that have a requirement. Some limited requirements can be met for some CDAs through the following contract vehicles:

- DLA contract with Storage Technology
- Army Amdahl contract - AIPCs only

These contract vehicles and the CDAs that could potentially access them are described in section 2.3.1.

Acquisition of disk drives, tape drives, add-on memory, controllers, and other IBM mainframe-compatible peripheral equipment is not possible through 8(a) contract vehicles unless other services such as maintenance, training, and system analysis support are required that allow the 8(a) contractor to provide over 50 percent of the contract value directly. In other words, the peripheral equipment must be less than 50 percent of the contract value if the contractor is not the manufacturer. In addition, the contractor must meet the requirements of the Walsh-Healey Act. If the CDA has a requirement to purchase mainframe equipment, peripheral equipment can then be purchased in conjunction with the mainframe using the SIC 5045 exclusion rules of SBA.

2.3.1 Existing Contract Vehicles for Peripherals

Three contract vehicles are currently feasible for meeting some CDA requirements:

• Storage Technology	DLA Contract
• Amdahl Contract	Army IPC Contract
• DISA Omnibus Contract	Advance Corporation

A summary of each contract is provided later in this appendix. None of three vehicles supports DISA wide requirements. The DLA and Army contracts are constrained in terms of who can purchase from the contract and the DISA contract requires that the hardware purchase be accompanied by at least a 50 percent services content.

2.3.2 8(a) Contracting Possibilities for Peripherals

Contracting for peripherals through the SBA 8(a) program does not appear to be possible given the 51 percent content constraint. Peripherals can, however, be part of a mainframe equipment acquisition under the SIC 5045 exception discussed in section 2.2.1.

2.4 Communications Equipment

2.4.1 Front-End Processors

Two potential sources of front-end processors, communications controllers, and display systems for bisynchronous and SNA networks are as follows:

- NCR/Comten Corporation
- IBM

At present we have not identified a contract vehicle other than the IBM GSA schedule that supports the acquisition of this equipment. In those cases where a 50 percent services content is acceptable, the DISA Omnibus contract may be applicable. Some Marine Corps requirements may be satisfied through the Marine Corps Front-End Processor (MCFEP) contract summarized later in this appendix.

2.4.2 DISN Gateway Equipment

The IBM mainframe to be accessed by logistics users through DISN using GOSIP protocols must be connected to the DISN using a gateway device. The most common approach to this connectivity is to connect the IBM mainframe to an Ethernet LAN using an Ethernet-to-IBM mainframe gateway. The Ethernet LAN is then in turn connected to DISN through an Ethernet-to-DISN gateway. Companies that manufacture IBM-to-Ethernet/TCP/IP gateways are Fibronics and ACC. Whichever product is proposed, it must support a complete TCP/IP socket interface to the IBM mainframe to allow process- to-process communications in addition to TELNET and File Transfer services.

The following are potential contract vehicles for acquisition of IBM-to-Ethernet gateway equipment:

• IBM GSA Schedule	TCP/IP for MVS, Part No. 5685-061
• AFCAC 300	Interlink 3722 and Access MVS
• Small Purchase	Fibronics K-Net product

Of these potential contract vehicles, all appear usable. However, the AFCAC 300 vehicle is the least cumbersome and will probably be capable of supporting higher purchase quantities.

Section 3 - User Access Area Equipment

3.1 Introduction

Equipment required in the user access area consists of UNIX workstations and servers, X Terminals, and personal computers using UNIX or functioning as X Terminals. In addition, the user access area includes communications equipment to interconnect user workstations and file servers, and to support access to remote sites through the Defense Information System Network. Communications devices consist of LAN equipment for both simple Ethernet LANs and for high-bandwidth fiber optic LANs with Ethernet access and communications gateways to support Ethernet LAN-to-DISN access.

3.2 Communications Equipment

Both ID/IQ contracts and GSA schedules exist for acquiring Ethernet LAN components for installation of user access area LANs. The best vehicle for acquisition of Ethernet LAN components appears to be the PC-LAN contract managed by the Navy. This contract is an ID/IQ contract with Ethernet cables, transceivers, and connectors currently included. For systems to be implemented later in 1993, the ULANA II contract appears to be the best acquisition vehicle. It is currently being purchased and is expected to be awarded in late 1993.

The PC-LAN and ULANA II contracts are described in more detail in the Communications Equipment Contract Vehicle Descriptions section. Those contracts and GSA schedules that can be used to acquire communications components are as follows:

- PC-LAN Contract - page A-53
- Small Multiuser Computer (SMC) Contract - page A-35
- GTSI GSA Schedule - page A-44
- EDS ULANA Contract - page A-54
- ULANA II Contract - page A-55

3.2.1 FDDI Backbone LANs

In cases where a significant number of X Terminals are to communicate with a single UNIX server, the capacity of an Ethernet LAN may be insufficient. In these cases, FDDI backbone LANs or other higher speed LANs are potentially required. Contract vehicles supporting acquisition of high-speed LAN backbones with Ethernet interfaces are
as follows:

•	Fibronics	PC-LAN
•	Cisco Systems	GSA Schedule

3.2.2 Ethernet LANs

Components required to construct Ethernet LANs include transceivers, cable, connectors, fanout boxes, bridges, and interface cards. Sources of these components are as follows:

• Transceivers	PC-LAN, SMC, GTSI
• Cable	SMC, PC-LAN, GTSI
• Connectors	PC-LAN, SMC, GTSI
• Fanout Boxes	PC-LAN, SMC, GTSI
• Interface Cards	PC-LAN, SMC, GTSI

3.2.3 DISN Gateways

Ethernet-to-DISN gateways are provided by Cisco Systems. The Cisco Systems product is available through the Cisco Systems GSA schedule and the ULANA and ULANA II contracts.

3.3 UNIX Servers

In those cases where UNIX servers are required to support X Terminal users, this equipment can be acquired through the contract vehicles listed below. The NPIC contract is currently being renegotiated and should be available by 1 October. Once awarded, the AFCAC 300 contract appears to be the most advantageous.

• Sun Servers	Sylvest GSA, NPIC
• HP Servers	Sylvest GSA
• UNIX Servers	AFCAC 300

3.4 UNIX Workstations

Potential sources for UNIX workstations are Sun Microsystems and Hewlett-Packard (HP). Both products are included in the Sylvest Systems GSA schedule. The NPIC contract provides a further vehicle for Sun workstations. The most advantageous source, however, appears to be the AFCAC 300 contract (once awarded).

• Sun	NPIC, Sylvest GSA
• HP Workstations	Sylvest GSA
• UNIX Workstations	AFCAC 300

3.5 X Terminals

X Terminals are currently manufactured by numerous companies including HP, Tektronics, Human Design Systems, Network Computing Devices (NCD), and Visual. At present the most advantageous contract vehicle for this equipment is the AFCAC 300 acquisition. A longer term potential vehicle is the Air Force SEWS acquisition, which is just now being initiated. If DISA/TIM were to aggressively pursue adding their X Terminal and workstation requirements to this acquisition, a long-term vehicle would potentially be available in about a year to 18 months. Once awarded, the AFCAC 300 contract is, however, the best current source.

- HP X Terminals
- Tektronics X Terminals
- Human Design Systems (HDS)
- NCD
- Visual

3.6 Personal Computers and Associated Items

Those personal computer-related components required to support the DISA/TIM logistics workstation architecture are PC hardware and DOS applications including X-Server applications running under either DOS or MS Windows on the PC.

3.6.1 Personal Computers

A number of contract vehicles exist for acquisition of personal computers and associated software packages. The most advantageous for DISA will be the Desktop IV contract once the current protest is resolved. Current contract vehicles are as follows:

• PCs 486	SMC, Desktop IV, DECCO
• PCs 386	SMC, DECCO
• PCs 386/486	GSA Schedule GTSI

3.6.2 X Window Servers

X Window servers for PCs using DOS and/or Microsoft Windows are available from the following companies:

- Hummingbird Communications
- PC X-View

At present, no contract vehicle or GSA schedule has been identified that could be used to purchase these software packages. The license fee on a per-copy basis is low enough that the small purchase authorization of each CDA should allow these items to be purchased.

Mainframe Contract Vehicle Descriptions

Contract Vehicle #1

Contract Vehicle:	DISA Omnibus
Government Contact:	John Godbey- CO (703) 692-2782 Rhonda LaGard- Contract Specialist (703) 692-3706
Contractor Contact:	Advance Corporation MelvinLaney (202) 785-9220
Scope:	All DISA activities including CIM initiatives.
Equipment Provided:	IBM compatible mainframes, peripherals, and communications equipment.
Contract Summary:	This contract can be used to acquire IBM compatible mainframe, peripheral, or communications equipment. Most other ADP equipment can also be purchased through this vehicle. The primary constraint is that the contractor must maintain over 50 percent content in any order placed under this contract. That means that the dollar volume of hardware ordered under this contract must be less than the dollar volume of services. Services can include installation, operation, training, systems analysis, etc. Up to $30 million of hardware and services can be ordered under this contract vehicle commencing 1 October 1992.

Contract Vehicle #2

Contract Vehicle:	GSA Drop-In Technology
Government Contact:	Ann Liddle - GSA (703) 756-4500
Contractor Contact:	N/A
Scope:	Still to be determined but at a minimum will include all of DoD.
Equipment Provided:	IBM mainframe equipment. No peripherals are included in this contract vehicle.
Contract Summary:	GSA has initiated an ID/IQ acquisition for IBM compatible mainframe equipment. An initial CBD announcement is expected in mid-October 1992 with final award of a contract in late 1993. If this acquisition goes well, GSA may initiate an acquisition for peripherals in 1993.

Contract Vehicle #3

Contract Vehicle:

AIPC DASD
Contract No. DAHC94-92-D-0007

Government Contact:

Gloria McGee
Contracting Officer
U.S. Army ISSA
(703) 325-3334

Contractor Contact:

Iryna Yasinska
Sr. Contract Administrator
(202) 895-4317

Scope:

All Army Information Processing Centers.

Equipment Provided:

Amdahl DASD, DASD controllers, cache memory, and channel adapters (Technology insertion clause may permit support of other direct access storage devices.)

Contract Summary:

This ID/IQ contract, in place until May 1997, allows any Army Information Processing Center to order additional direct access storage devices for existing Amdahl mainframes, the associated controllers and channel adapters, and any required cache memory for the desired I/O throughput.

Contract Vehicle #4

Contract Vehicle: 8(a) 5045 Contract

Government Contact:
Neil Wensel
SBA Washington District Office
(202) 634-1500 x 341

Contractor Contact:
Mr. Chuck Yarborough
5045 Program Director
Government Micro Resources
14121 Parke Long Court Unit 104
Chantilly, VA 22021
Telephone Voice (703) 263-9149

Scope: All Department of Defense.

Equipment Provided: IBM and IBM compatible mainframes, peripherals, and communications equipment; associated software.

Contract Summary: GMR is one of three IBM-certified dealers. The company specializes in product sales and services. It is well known as a dealer in microcomputers but has recently expanded into departmental and mainframe systems. It was certified in the 8(a) program in 1987. The company has provided mainframes and associated peripheral equipment under a similar contract vehicle to NASA and the Department of the Treasury.

The company holds an FY 1993 GSA schedule.

Contract Vehicle #5

Contract Vehicle: 8(a) 5045 Contract

Government Contact:
Neil Wensel
SBA District Office
(202) 634-1500 x 341

Contractor Contact:
Mr. Wayne Carter
President
D.C. Information Systems
8121 Georgia Avenue, Suite 600
Silver Spring, MD 20910
Telephone - Voice (301) 585-6103
Facsimile (301) 585-4128

Scope: All Department of Defense.

Equipment Provided: IBM and IBM compatible mainframes, peripherals, and communications equipment; associated software.

Contract Summary: DCIS is one of three IBM-certified dealers. The company specializes in mainframe product sales and services. It was certified in the 8(a) program in 1989. The company has provided mainframes and associated peripheral equipment under a similar contract vehicle to Smithsonian Institution and the Administrative Office of the U.S. Courts.

The company does not hold a GSA schedule.

Contract Vehicle #6

Contract Vehicle: 8(a) 5045 Mainframe Waivers

Government Contact:
Charita Albright
Business Opportunity Specialist
SBA Washington District Office
(202) 634-1500 X307

Contractor Contact:
Mr. Marcus Price
President
Technical Software Services, Inc.
1400 Mercantile Lane
Landover, MD 20785
Telephone - Voice (301) 772-0101
Facsimile (301) 772-0111

Scope: All Department of Defense.

Equipment Provided: IBM and IBM compatible mainframes, peripherals, and communications equipment; associated software.

Contract Summary: The Small Business Administration has waived the nonmanufacturer rule for mainframe computers and associated peripherals and communications equipment since no small business manufacturer is available to participate in the Federal market for this class of products. This waiver allows an agency to acquire this class of products from an otherwise qualified 8(a) regular dealer through the SBA 8(a) set-aside program. The procuring agency is required to make the determination that the 8(a) contractor is a regular dealer in accordance with the Walsh-Healey Act. The contracting office of the procuring agency has the responsibility to assure price reasonableness. For sole-source contracts, the timeframes being experienced in mainframe set-asides range from 30 to 90 days. Competitive 8(a) procurements may take up to 180 days longer.

This contractual vehicle requires a minimum guaranteed contract amount at the time of initiation. TSSI was certified as an 8(a) company (with 5045 as a primary

Contract Vehicle #6 (cont'd)

Standard Industrial classification code, among others) in 1991. This contract vehicle will therefore exist until the year 2000. TSSI's key corporate strength is its focus on mainframes. In addition to providing mainframe equipment and software, the company offers expertise in applications systems development and general technical support capabilities in the mainframe area.

Contract Vehicle #7

Contract Vehicle: U.S. Marine Corps CPU Augmentation
Contract No. N66032-90-D-0008

Government Contact: Kenny R.Bluteau (COR)
Nancy Flake, Buyer
Headquarters, USMC
(703) 696-1011

Contractor Contact: Steve Jacek
Director, Business Development
Severn Companies, Inc.
(301) 306-4115

Scope: U.S. Marine Corps for the augmentation of currently existing USMC Amdahl systems in USMC computer facilities, CONUS and OCONUS.

Equipment Provided: Amdahl processors (5890, 5995), main memory augmentations, channel adapters, Amdahl's UTS operating system, Multiple Domain Feature, MLS Software.

Contract Summary: Equipment can be ordered under this contract until 30 September 1994. System software other than the Amdahl software listed above is available through the applicable IBM GSA schedule or other contractual vehicles. Up to 14 processors, of which 5 have already been ordered and installed, can be acquired off this contract. No maximum is prescribed for other devices and software provided the equipment will be installed in the mainframe systems procured through this contract.

Contract Vehicle #8

Contract Vehicle:	MCFEP Contract No. N66032-92 -D-0008
Government Contact :	Cathy Ziegler Contracting Officer U.S. Marine Corps 3033 Wilson Blvd., Suite 722 Arlington, VA 22201 (703) 696-1010/1014
Contractor Contact:	Bob Sturm Severn Companies, Inc. (301) 794-9680
Scope:	Marine Corps, mostly CONUS.
Equipment Provided:	Idea Courier 10300 communications controllers, Ethernet or Token Ring adapters, warranties, maintenance, and training.
Contract Summary:	This contract is in place until 9 June 2000 and allows any Marine Corps unit to purchase listed products and services. Up to 1,556 communications controllers, 717 Token Ring interfaces, and 284 Ethernet interfaces. Under the MCFEP contract, technology improvements may be proposed "to save money," to improve performance, to save energy, or for any other purpose that presents a technological advantage to the Government.

Contract Vehicle #9

Contract Vehicle: DLA DASD and Tape Library

Government Contact: Phil Silas
(703) 274-3210

Contractor Contact: Storage Technology
Dennis Lucey
(301) 680-1337

Scope: DLA sites with existing IBM compatible systems.

Equipment Provided: Storage Technology DASD and tape libraries.

Contract Summary: This contract was intended to provide peripherals for DLA systems. Some possibility exists that other DISA requirements can be met through this contract vehicle. The primary issue is contract scope. Sufficient quantities of DASD and tape library CLINs are available to meet both DLA and other requirements.

UNIX Workstation Contract Vehicle Descriptions

Contract Description #1

Contract Vehicle:	Small Multiuser Computer (SMC)
Government Contact:	Mary Morris Defense Logistics Agency- ZIA Building 3 Cameron Station DSN: 284-7506/8 Comm: (703) 274-7506/8
Contractor Contact:	Electronic Data Systems Sales Ann Haller (703) 742-1585 Support (800) 762-3371
Scope:	Army, Navy, Marine Corps, Defense Logistics Agency, Reserves, and Coast Guard.
Equipment Provided:	Personal computers (UNIX/DOS) File servers (UNIX) LAN cards for PCs and file server Informix 4.0/4.1 (SQL RDBMS)
Available Documentation:	CLINs and CLIN descriptions.

Contract Description #2

Identifier:	Desktop IV
Government Contact:	Maj.J.D. Smith Gunter AFB (205) 416-3464
Contractor Contact:	Unknown; acquisition being protested.
Scope:	All services and DoD agencies. Contract awarded but was protested as of 21 October 1992.
Equipment Provided:	PC hardware and software including UNIX OS and related applications to run on PCs.
Contract Summary:	N/A

Contract Description #3

Identifier:	TAC III
Government Contact:	Mary AnnGroomes (202) 433-2387
Contractor Contact:	Hughes (800) 843-1101
Scope:	Limited to requirements for tactical systems. Therefore, not usable for DISA/TIM NTI requirements.
Equipment Provided:	HP workstations (UNIX) HP servers (UNIX and X Window)
Contract Summary:	N/A

Contract Description #4

Identifier:	Army Reserve Component Automation System (RCAS)
Government Contact:	Army PM Mr. Tompkins - Deputy PM (703) 339-1741
Contractor Contact:	Boeing
Scope:	RCAS installations only. No provision for purchases of equipment only.
Equipment Provided:	HDS X Terminals
Contract Summary:	N/A

Contract Description #5

Identifier:	NPIC
Government Contact/PM:	Grant Chisolm (202) 863-3750
Contractor Contact:	Sun Micro Systems Lisa Wells (513) 254-9070
Scope:	All services and DoD agencies. Currently being renegotiated. Some possibility exists that the scope of this contract may be limited to intelligence agencies commencing Fiscal Year 1993.
Equipment provided:	Sun workstations (UNIX) Sun servers (UNIX and X Window)
Contract Summary:	N/A

Contract Vehicle #6

Contract Vehicle:	AFCAC 308
Government Contact:	Mrs. Durette (804) 764-4503
Contractor Contact:	N/A
Scope:	Specific Air Force programs.
Equipment Provided:	UNIX workstations
Contract Summary:	This contract provides UNIX workstations for DoD. The contract quantities are depleted except for reserved quantities for specific Air Force programs.

Contract Description #7

Identifier:	JCALS
Government Contact:	Army PM Jim Tomaldson (908) 532-0400
Contractor Contact:	Computer Sciences Corporation Morristown, NJ Bud Caldwell- PM Lew Rebecca- Deputy PM (609) 234-1100
Scope:	JCALS installations only. No provisions for equipment-only purchases.
Equipment Provided:	X Terminals.
Contract Summary:	N/A

Contract Description #8

Identifier:	DECCO Computer Store
Government Contact:	DECCO-Scott AFB (618) 256-9690 Bulletin Board - (618) 744 -8787
Contractor Contact:	Various.
Scope:	All services and agencies within DoD.
Equipment Provided:	Personal computers File servers PC LANs
Contract Summary:	N/A

Contract Vehicle #9

Contract Vehicle:	Standard Engineering Wokstation (SEWS)
Government Contact:	Barry Cain (513) 257-6721 Chuck Schneggenburger (513) 255-3366
Contractor Contact:	N/A
Scope:	To be determined.
Equipment Provided:	Engineering workstations and potentially X Terminals, UNIX servers, etc.
Contract Summary:	This acquisition is in a preliminary stage with no anticipated date for RFP release.

Contract Vehicle #10

Contract Vehicle:	GTSI GSA Schedule
Government Contact:	GSA Schedule
Contractor Contact:	GTSI Sales (800) 999-4874
Scope:	All Government agencies and departments.
Equipment Provided:	Primarily personal computers and related local area networks. Also includes UNIX workstations and X Terminals.
Contract Summary:	This is a large list of PC hardware and related software and communications equipment.

Contract Vehicle #11

Contract Vehicle:	Government Micro Resources- GSA Schedule
Government Contact:	GSA Schedule
Contractor Contact:	GMR Sales Office (800) 232-4671
Scope:	All Government agencies and departments.
Equipment Provided:	Primarily personal computer hardware and software but also includes LAN equipment, printers, etc.
Contract Summary:	Comprehensive GSA schedule for office automation applications.

Contract Vehicle #12

Contract Vehicle:	HFSI GSA Schedule
Government Contact:	GSA Schedule
Contractor Contact:	Elizabeth Fox HFSI (703) 827-3160
Scope:	All Government departments and agencies.
Equipment Provided:	UNIX workstation add-in cards for personal computers.
Contract Summary:	GSA schedule.

Contract Vehicle #13

Contract Vehicle:	Sylvest GSA Schedule
Government Contact:	GSA Schedule
Contractor Contact:	Maribeth Girard (301) 459-2700
Scope:	All Government agencies and departments.
Equipment Provided:	UNIX workstations, X Terminals, and related software.
Contract Summary:	GSA schedule.

Contract Vehicle #14

Contract Vehicle:	NETCAP
Government Contact:	Maj. Schaeffer (617) 271-8848
Contractor Contact:	N/A
Scope:	Limited to intelligence community users IDHS and DoDIIS.
Equipment Provided:	X Terminals.
Contract Summary:	This contract includes X Terminals but the scope is limited to IDHS and DoDIIS users.

Contract Vehicle #15

Contract Vehicle:	AFCAC 300
Government Contact:	Capt. Brown- AFCAC (617) 377-8638
Contractor Contact:	N/A
Scope:	Department of Defense.
Equipment Provided:	UNIX servers, UNIX workstations, X Terminals, LANs, and related software.
Contract Summary:	ID/IQ contract due to be awarded in mid-October 1992. This appears to be the must likely vehicle for X Terminal, UNIX workstation, and LAN requirements for NTI requirements.

Contract Vehicle #16

Contract Vehicle:	AFCAC 305
Government Contact:	Capt. Paul M. Commeau (617) 377-8638
Contractor Contact:	N/A
Scope:	Services, DLA, DISA, IRS, and other Federal agencies.
Equipment Provided:	UNIX-based database machines.
Contract Summary:	Acquisition of database machines for installation at sites with host computers requiring relational database services.

Contract Vehicle #17

Contract Vehicle: WIS Workstation

Government Contact: Joseph G. Cloutier, WIS PM
Gunter, AFB Alabama

Contractor Contact: HFSI
Bud Smith - PM
(703) 827-3533
John Krasula
(703) 827-3718

Scope: All Government departments that are associated with Command and Control.

Equipment Provided: Macintosh workstation with AIX (UNIX) operating system and associated software and peripherals. Macintosh FX personal workstations, processor and memory augmentations, various peripherals, A/UX with security enhancements, personal workstation software for word processing/spreadsheets, business graphics, DBMS, LAN boards, Amdahl processors, main memory augmentations, channel adapters, UTS operating system.

Contract Summary: This contract allows authorized DoD units to order personal workstation equipment and software.

Communications Equipment Contract Vehicle Descriptions

Contract Description #1

Identifier:	PC LAN
Contract Number:	F-19630-91 -D-0001 (AFCAC)
Government Contact/PM:	Jakki Rightmeyer Shirley Dumbar NCTAMS LANT, Code N811.2 138 E. Little Creek Road Norfolk, VA 23502-2551 Comm: (804) 445-1493
Contractor Contact:	Digital Equipment Corporation (800) Navy LAN [(800) 628-9526]
Scope:	All services and DoD agencies.
Equipment Provided:	Personal computers (DOS) PC LAN OS and interface cards (Novell) Ethernet LAN cables, connectors, and transceivers Attempting to add GOSIP LAN interface cards and interface software. Potentially available late 1992.
Contract Summary:	N/A

Contract Vehicle #2

Contract Vehicle:	ULANA
Government Contact:	N/A
Contractor Contact:	Dusty Wince (703) 742-2406
Scope:	All Government agencies and departments.
Equipment Provided:	Various LAN and gateway equipment.
Contract Summary:	Supports acquisition of communications equipment and LANs for DoD. Purchases limited to $25,000 under this basic ordering agreement.

Contract Vehicle #3

Contract Vehicle:	ULANA II
Government Contact:	Lt. King Tinker AFB (405) 734-9773 WendelCrittenden (405) 734-9928
Contractor Contact:	N/A
Scope:	All Department of Defense.
Equipment Provided:	Communications equipment, LANs, and gateways.
Contract Summary:	This acquisition is now in process with award expected in late 1993.

Contract Vehicle #4

Contract Vehicle:	AFCAC 300
Government Contact:	Capt. Brown- AFCAC (617) 377-8638
Contractor Contact:	N/A
Scope:	Department of Defense.
Equipment Provided:	UNIX servers, UNIX workstations, X Terminals, LANs, and related software.
Contract Summary:	ID/IQ contract due to be awarded in mid-October 1992. This appears to be the most likely vehicle for X Terminal, UNIX workstation, and LAN requirements.